

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR



INSTALACIÓN DE ROBOT PALETIZADO



PROYECTO FIN DE CARRERA

INGENIERÍA TÉCNICA INDUSTRIAL
ELECTRÓNICA INDUSTRIAL

AUTOR: JUAN CARLOS BLANCO FUENTES



DIRECTOR/ES: DMITRY KAINOV
JAUME GESS LÁZARO

Septiembre 2009



	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA Instalación de Robot Paletizado.	

INDICE



PROYECTO FIN DE CARRERA.....	1
1 INTRODUCCIÓN “Piensos O ´ Couto”.....	7
1.1 PRESENTACIÓN DEL PROBLEMA.....	7
1.2 OBJETIVOS DEL PROYECTO.....	7
1.3 PERSPECTIVAS FINALES Conclusiones_1.....	9
1.4 LA PALETIZACIÓN.....	11
1.4.1 Ventajas.....	12
1.4.2 Unidades de carga y aspectos constructivos del PALET.....	13
1.5 KUKA SYSTEM.....	15
1.5.1 El kr 180-2 pa.....	17
1.5.2 Campo de Aplicaciones (Paletización).....	17
2 SINOPTICO DE LA INSTALACIÓN.....	19
3 COMPONENTES HARDWARE (Paletizado).....	27
3.1 HERRAMIENTA DEL ROBOT (Garra).....	27
3.2 SEGURIDADES GENERALES (Barreras protectoras).....	32
3.2.1 Rearme de las seguridades del robot.....	33
3.2.2 Rearme de las barreras de seguridad.....	33
3.3 ARMARIO ELECTRICO. (Magelis XBT-N401).....	37
3.3.1 Descripción de componentes.....	37
3.3.2 Cable CAN-OPEN.....	41
3.3.3 Autómata TSX 3710.....	42
3.3.3.1 Conector terminal RS 485.....	42
3.3.3.2 Características del TSX 37-10.....	43
3.3.3.3 Bloque de visualización en el TSX 37-10.....	44
3.3.3.4 Mini-rack de extensión.....	45
3.3.3.5 Conexión de las alimentaciones.....	46
3.3.4 Descripción de mando.....	46
3.3.5 Terminal de diálogo XBT-N401.....	50
3.3.5.1 Menú Principal.....	53
3.3.5.2 Página LINEA 1.....	54
3.3.5.3 Página Datos de producción.....	58
3.3.5.4 Página Ordenes Robot.....	59
3.3.5.5 Página Mandos Manuales.....	65
3.3.5.6 Página de Alarmas.....	66
3.3.5.7 Páginas de Sistema.....	72
4 SISTEMA ROBOT KUKA.....	74
4.1 OBJETIVO DEL DOCUMENTO.....	74
4.2 ELEMENTOS BASICOS DE UN SISTEMA ROBOT.....	75
4.2.1 Armario de Control KR C2.....	75
4.2.1.1 PC de Control.....	77
4.2.1.2 Unidad de alimentación KPS-600.....	77
4.2.1.3 Acumuladores.....	78

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA Instalación de Robot Paletizado.	

4.2.1.4	Servo convertidores KSD1.	78
4.2.1.5	Conector (X11).	78
4.2.2	KUKA Control Panel (KCP).	80
4.2.2.1	Selector de Modos de servicio (1).	80
4.2.2.2	Accionamiento en ON/OFF (2)(3).	81
4.2.2.3	Pulsador Seta de emergencia (4).	81
4.2.2.4	Teclas de Arranque y parada (12), (13) y (14).	81
4.2.2.5	Tecla de Validación (7) y cambio de ventana (15).	81
4.2.2.6	Teclas de estado derecha (6), (17), (18) y de función (11).	82
4.2.2.7	Space-Mouse (Ratón espacial) (5).	82
4.2.2.8	Pulsador de Hombre Muerto (3) Parte Trasera.	83
4.2.3	Tarjeta electrónica (ESC).	83
4.3	ARRANQUE DE LA UNIDAD DE CONTROL.	84
4.4	SISTEMA DE COORDENADAS.	85
4.4.1	Movimientos Manuales.	85
4.4.1.1	Sistema de coordenadas específico eje a eje.	86
4.4.1.2	Sistema de coordenadas en WORLD.	86
4.4.1.3	Orientación de la herramienta STATUS "S" TURN "T".	88
4.4.2	Medición de una base.	88
4.5	MOVIMIENTOS DEL ROBOT.	89
4.5.1	Punto de trayectoria alcanzada (COI).	90
4.5.2	Movimiento (PTP).	91
4.5.2.1	Programación de un movimiento PTP.	91
4.5.2.2	Programación de un movimiento LIN.	94
4.5.2.3	Programación de un movimiento CIRC.	95
4.5.3	Concepto del Advance.	96
4.5.4	Ejecutar un programa.	97
4.5.4.1	Velocidad de Programa.	97
4.5.4.2	Modos de ejecución de un programa.	97
4.5.4.3	La barra de estados (Interpretador SUBMIT).	98
4.5.4.4	Desplazamiento de coincidencia COI.	100
4.5.5	Drivers E/S.	100
4.5.6	Menú Indicación.	102
4.5.7	Entradas y salidas digitales.	103
4.6	PROGRAMACIÓN LÓGICA.	105
4.6.1	Grupos de Usuarios.	105
4.6.2	Menu Instrucción.	105
4.6.3	Estructura de un programa.	106
4.6.4	Declaración de Variables.	106
4.6.4.1	Variable entera "INT".	107
4.6.4.2	Variable real/booleana/carácter "REAL/BOOL/CHAR"	107
4.6.4.3	Variable de tipo Estructura "STRUC".	107
4.6.4.4	Variables "SIGNAL".	108
4.6.4.5	Variables de sistema.	108
4.6.5	OPERADORES.	108



	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

4.6.5.1	Operadores aritméticos.	108
4.6.5.2	Operador Geométrico ":"	108
4.6.5.3	Operadores de comparación.....	109
4.6.5.4	Operadores lógicos.	109
4.6.6	Instrucciones de Programa.....	109
4.6.6.1	Rama Condicional "IF / ENDIF".....	109
4.6.6.2	Distribuidor "SWITCH / ENDSWITCH".....	110
4.6.6.3	Bucles sin fin "LOOP / ENDLOOP".	110
4.6.6.4	Instrucción "HALT".....	110
5	AUTOMATICO EXTERNO (PLC).....	111
5.1	SEÑALES DE ENTRADAS.	112
5.2	SEÑALES DE SALIDA.	114
5.3	DIAGRAMA DE SEÑALES.	115
6	SOFTWARE DE LA INSTALACIÓN.	117
6.1	CONFIGURACIÓN DE MOSAICOS.....	117
6.1.1	Configuración de tres sacos.....	120
6.1.2	Configuración de cuatro sacos.	121
6.1.3	Configuración de cinco sacos.	122
6.2	UN PROGRAMA DE ROBOT.	123
6.2.1	Variables Físicas.	123
6.2.1.1	Entradas.	123
6.2.1.2	Salidas.	132
6.2.2	Variables declaradas.	134
6.2.2.1	Base_data [].	138
6.2.2.2	Otras Variables.	140
6.2.3	Diagramasgramas de estado software del robot.....	142
6.2.3.1	Interpretador Submit (Multitarea SUB.sps).	142
6.2.3.2	Automático externo (CELL.src).	144
6.2.3.3	Home.src.....	146
6.2.3.4	Coi_sgrd.src.	148
6.2.3.5	Mantenimiento. src	150
6.2.3.6	Mosaico_3. src / Mosaico_4.src / Mosaico_5.src	152
6.2.3.7	Zona_1.src / Zona_2.src	158
6.2.3.8	Zona_11.src / Zona_22.src.....	164
6.2.3.9	Zona_111.src /Zona_222.src	167
6.2.3.10	Coger_1.src / Coger_2.src.....	172
6.3	PROGRAMA TSX 3710.....	173
6.3.1	Entradas digitales:	173
6.3.2	Comunicación con el robot.	178
6.3.2.1	Diagrama de estados – Manejo del robot.	178
6.3.3	Salidas digitales.....	182
6.3.4	Comunicación con terminal XBT.....	186
7	CONCLUSIONES.....	194
8	ANEXO.A: (Programa multitarea Sps.sub).....	197
9	ANEXO.B: (Programa Automático externo).	198

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA Instalación de Robot Paletizado.	

9.1	"CELL.SRC"	198
10	ANEXO.C: (Esquema eléctrico de la instalación).....	199
11	ANEXO.D: (Programa del PLC).....	213
11.1	VARIABLES.	213
11.1.1	Bit(s) Internos(s).....	213
11.1.2	Palabra(s) Internas(s).	213
11.1.3	Temporizador(es).	215
11.1.4	Módulo de entradas_1.	215
11.1.5	Módulo de entradas_3.	216
11.1.6	Módulo de entradas_5.	216
11.1.7	Módulo de salidas_2.....	216
11.1.8	Módulo de salidas_4.....	217
11.1.9	Módulo de salidas_6.....	217
11.2	PROGRAMACIÓN EN CONTACTOS.....	218
11.2.1	LD_Mast_General.	218
11.2.2	LD_Mast_Robot_Auto_Extern.	219
11.2.3	LD_Mast_Recogida_De_Saco.....	227
11.2.4	LD_Mast_Saturación.	231
11.2.5	LD_Mast_Permisos.....	232
11.2.6	LD_Mast_Com_Robot.	234
11.2.7	LD_Mast_Alarmas.....	251
11.2.8	LD_Mast_Producción.	254
12	ANEXO.E: (Programa de Robot Paletizado).	255
12.1	MOSAICO_3.src / MOSAICO_4.src. / MOSAICO_5.src.	255
12.2	MANTENIMIENTO.src.....	256
12.3	HOME.src.....	257
12.4	COI_SGRD.src.....	257
12.5	ZONA_1.src / ZONA_2.src.....	258
12.6	ZONA_11.src / ZONA_22-src.	260
12.7	ZONA_111.src /ZONA_222.src.....	262
12.8	COGER_1.src.....	265
12.9	COGER_2.src.....	265
12.10	CERRAR.src.....	265



	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

1 INTRODUCCIÓN "Pensos O´Couto".

El Proyecto de fin de carrera pretendido, definirá una de las muchas herramientas en el mundo de la logística como son los sistemas de paletizado. De los muchos sistemas industriales que existen actualmente en el mercado se escogerá un sistema robotizado basado en los equipos **KRC-2 (KUKA-System)**, objeto principal de desarrollo de este proyecto, y más en concreto el robot de cuatro ejes para paletizar denominado **KR 180-2 PA**, este así mismo, estará integrado en la periferia de control mediante **PLC (Telemecanique)** en una de las empresas de alimentación para las que esta destinado este tipo de instalaciones, siendo el autómatas empleado el modelo **TSX371028DTK** de las industrias **"Sneider Electric"**.

El proceso consistirá en la paletización en europalets, de sacos de contenido agroalimentario, sector al que se dedica **"PIENSOS O´COUTO"** también conocido como **"Joaquín Valledor, SL"** (en honor al fundador) y que se encuentra en la localidad de A Fonsagrada (Lugo). Estos fabricantes de piensos animales, realizan actualmente su actividad comercial y prestan sus servicios de Agricultura y Ganadería.

1.1 PRESENTACIÓN DEL PROBLEMA.



"Pensos O´Couto" muy lejos de poseer unas instalaciones modernizadas; cosa que es habitual en fabricantes de menor escala, empleaba para paletizar su producto final la mano de obra propiamente dicha, siendo el ser humano el encargado de coger cada uno de los sacos que salen de la línea de fabricación y depositarlos manualmente en el palet para su embalaje final. Este resultaba ser un método de trabajo que consumía grandes recursos humanos por el desgaste físico que resultaba ser para los operarios encargados de la paletización; necesitando por lo tanto, disponer de personal suficiente y distribuido por turnos para la ejecución del trabajo, y así cumplir con los objetivos y compromisos finales de producción de la empresa.

Ante el crecimiento productivo de la empresa, el sistema empleado resultó ser poco ortodoxo y rudimentario, ya que algunos de los pesos que se manejaban superaban los 40kg por saco.

1.2 OBJETIVOS DEL PROYECTO.

El objetivo de este proyecto consistirá en automatizar el sistema de paletizado usado, mediante el empleo de un brazo robotizado tipo **KUKA Kr 180-2 PA** destinado a la paletización y el autómatas **TSX 3710** de Telemecanique para el control periférico y comunicación con el robot mediante la Interface automático externo, donde el operario manejará la instalación, incluido el robot desde los elementos de mando integrados en el armario de control sin interferir en ningún momento con la consola de programación del armario robot KRC-2. Para ello se desarrollarán los siguientes puntos.

1. Diseño y desglose de los componentes que compondrán nuestra instalación **SINÓPTICO**. Estos son:



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

- a. Brazo robotizado KUKA KR 180-2 PA.
- b. Dos líneas de paletizado en el suelo y con palet, europalet.
- c. Armario KRC-2 Destinado a las conexiones y manejo del KR.
- d. Armario de control que integra el autómatas para el control periférico y KR con elementos de mando.
- e. Vallado periférico que habilita el acceso restringido al perímetro del robot.
- f. Seguridades electrónicas para el acceso mediante barreras de seguridad.
- g. Elementos neumáticos, para el control de la garra o mordaza del robot.
- h. Elementos de presencia, basado en fotocélulas reflex o palpadoras para la toma de decisiones.
- i. Transportadores ligeros empleados en la instalación para la entrada de sacos a la herramienta principal del robot.

2. Descripción y desarrollo de cada uno de los elementos hardware de la instalación; esto es:

- a. Descripción de las seguridades: vallado electrónico y periférico.
- b. Descripción de los elementos de mando integrados en el armario de control.
 - i. Marcha.
 - ii. Paro.
 - iii. Reset de alarmas.
 - iv. Restos.
 - v. Bloqueo de Línea_1
 - vi. Bloqueo de Línea_2.
- c. Descripción Hardware del autómatas TSX 3710.
- d. Descripción Hardware de la mordaza y garra del robot empleado para la recogida y dejada de sacos en los palet's.
- e. Descripción Hardware del módulo Beckhoff empleado para las conexión de señales y manejo del KR.
- f. Descripción hardware del terminal de diálogo **XBT N-401** empleado para la Interface con el operador, y desde el cual se podrán configurar algunas de las opciones siguientes:
 - i. Selección del programa de paletizado en función del mosaico.
 - 1. Mosaico de 5 sacos por altura.
 - 2. Mosaico de 4 sacos por altura.
 - 3. Mosaico de 3 sacos por altura.
 - ii. Configuración del número de alturas a paletizar, siendo el máximo de 10 alturas.
 - iii. Manejo manual de los transportadores de la instalación.
 - iv. Estado y diagnóstico de la averías de la instalación mediante mensajes emergentes.
- g. Se detallará la herramienta software empleada para la configuración de uno de estos terminales XBT siendo en nuestro caso el **Vigeo Designer Lite**.

3. Descripción Hardware de un sistema robot Kuka.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

- a. Armario KRC-2.
 - b. Consola de programación o KCP.
 - c. Sistema de coordenadas.
4. Se dará una breve introducción a la programación lógica de un sistema KRC-2 con su sintaxis propia.
 5. Desarrollo Software mediante diagramas de estado de los programas de robot.
 - a. Cell.src -> Programa para el manejo mediante automático externo.
 - b. Programas de paletizado principales.
 - i. Mosaico_5 ().
 - ii. Mosaico_4 ().
 - iii. Mosaico_3 ().
 - c. Explicación detallada de los procedimientos a seguir para el cambio de programa de paletizado.
 6. Desarrollo software de la periferia y control del robot, mediante lenguaje de programación en contactos y a través del software PL7 Junior.

1.3 PERSPECTIVAS FINALES Conclusiones_1.





(Fig.1.3-1) KR 180-2 PA.

La solución implantada; basada en el brazo robotizado **KUKA Kr 180-2 PA**, una vez terminada la instalación automatizada para el paletizado de los sacos, dará mayor rendimiento y dinamismo productivo, también aliviará de carga de trabajo, ya que algunos sacos son demasiado pesados para los operarios.

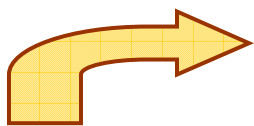
El KR-180 destinado a la paletización, se encargará de coger los sacos que salen de la línea de producción, mediante una mordaza estándar válida para los diferentes sacos en dimensiones geométricas con las que se trabaja en la obra, pudiendo ser capaz de manipular y paletizar una gran cantidad de productos diferentes, sacos de 25Kg., 30Kg. o 40Kg de todo tipo de producto ya sea material en grano o producto más filtrado como pueden ser las harinas.

En comparación con otras soluciones finales, el sistema de paletizado mediante brazo robot, resulta ser un elemento más barato y capaz de ser acoplado en espacios mínimos, no siendo tan restrictivo en el uso de espacios comparado con

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

otras maquinarias más pesadas; es por lo tanto, óptimo para empresas cuya producción no se ve

limitada por la ejecución de



movimientos del robot, y andan escasos de espacio; no obstante, es capaz de paletizar de entre 400-500 sacos la hora.



(Fig.1.3-3) Palet Terminado.



(Fig.1.3-2) Instalación de Paletizado.



Así mismo por el dinamismo de movimientos y la capacidad de alcance de los ejes, se conseguirá un punto de recogida adaptado a las infraestructuras de la nave, no siendo necesario adaptar mediante transportadores la llegada de los sacos.

El paletizado final que se pretende, ha de ser totalmente geométrico en las esquinas del palet, aunque esto dependerá de las características del producto y la homogeneidad con la que el saco llega a la línea de entrada del robot.

El sistema de agarre de sacos, permitirá la sujeción sin ejercer una presión sobre el mismo para no dañarlos, esto es debido a la doble acción de los dedos que sujetan el saco por debajo y las manos que lo harán lateralmente, además que incorpora una serie de sensores que determinará la posición de reposo de la garra de manera que nunca se pueda romper ningún saco en la recogida. Tras el paletizado, el producto final será sacado por una carretilla, y llevado a embalar para finalizar la sujeción y envase final que será transportado. Hay que decir que el recinto de trabajo se encontrará en todo momento cerrado al personal humano, posibilitando únicamente la entrada por medio de las barreras de seguridad que sólo se desactivaran una vez se haya terminado de paletizar sobre el palet para su recogida, y estando el robot fuera de la zona de recogida en ese momento.

Mediante este sistema, se conseguirá en caso de necesidad una disponibilidad del 100%; esto es, 24 horas de trabajo al día durante siete días a la semana con un personal mínimo, siendo capaz de paletizar más 16 toneladas de producto terminado a la hora, lo que hace un total de unas 400 o 500 toneladas al día.

La instalación no sólo estará controlada por el robot KUKA KRC-2, el cual podremos desplazarle mediante movimientos manuales a través de la consola **KCP**, si no que además, y una vez arrancado el programa automático del robot, este será controlado mediante los mandos de marcha y paro del armario eléctrico, siendo la puesta en marcha y cambio de programa de paletizado totalmente automático, con la mínima intervención del hombre al estar controlado por el PLC. En este sentido, los conocimientos que el operario debe adquirir sobre el manejo del robot, son mínimos, y su intervención solamente será necesaria para el cambio de mosaico

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

(dibujo del palet), o cada vez que haya que reiniciar el sistema en caso de avería o problema software, ya que el sistema KUKA se mueve bajo **WindowsXP**, y en ocasiones puntuales y bajo fallas de suministro eléctrico pueden quedarse bloqueados los sistemas internos o la comunicación con el PLC.

1.4 LA PALETIZACIÓN.

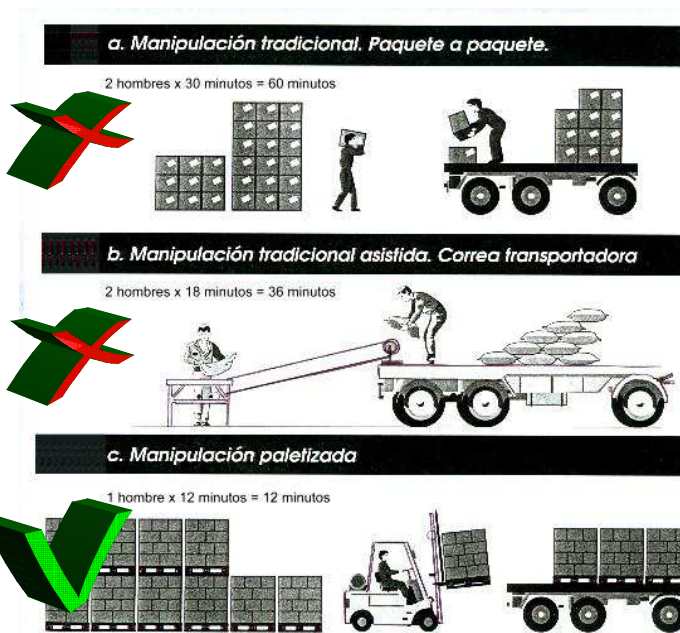
La paletización es sinónimo de almacenamiento y sirve para agrupar sobre una superficie plana (PALET, TARIMA o PALETA) cierta cantidad de material que en forma individual resulta poco manejable, pero que agrupándolo conforman una unidad de manejo para así poder trasladar, transportar y distribuir la mercancía final de forma segura, en poco tiempo y con el mínimo esfuerzo.

En la cadena logística interactúan las empresas industriales, los comerciantes y las "EPSL" (Empresas Prestadoras de Servicios Logísticos), entre los cuales existe un continuo flujo de materiales e información. Sin duda alguna, una de las operaciones más repetitivas en la cadena de distribución es la manipulación física de mercaderías. Siempre antes y después de un almacenamiento y de un transporte, por corto que este sea, existe una manipulación. La forma más lógica de reducir este costo es mecanizar las operaciones. Bajo estas condiciones, la mejor forma de reducir la manipulación es lograr mover de una sola vez el mayor número de cajas o productos en general. Esta es la razón de porque paletizar, ya que se logra una unidad de carga superior. La ventaja que acarrea pues, se encuentra en el ahorro de los sistemas de distribución, y la minimización en los tiempos de las operaciones de carga y descarga, ya que se producen menos manipulaciones en la mercancía. A parte está la optimización del espacio de almacenamiento al permitir mayores alturas del apilado, mejor aprovechamiento de los espacios de carga y vehículos, reducción del riesgo de roturas y pérdidas durante la manipulación y transporte.

La paletización ha sido considerada como una de las mejores prácticas dentro de los procesos logísticos, optimizando la eficiencia de los procesos que se realizan entre los integrantes de la cadena de abastecimiento.

1.4.1 Ventajas.

1. Aumento en la productividad.
2. Menor cantidad de mano de obra en las operaciones.
3. Disminución en los costos de carga y descarga.
4. Mejor aprovechamiento del espacio para almacenamiento sea en piso o en estantería.
5. Disminución de los tiempos de carga, descarga y almacenamiento.



6. Almacenamiento vertical con el uso de estanterías.



(Fig.1.4-2) Almacenamiento

7. Mejora los procesos de clasificación de productos en bodega.
8. Disminuye los daños de los productos al reducirse la manipulación.
9. Mejora la presentación de los productos favoreciendo la imagen de la marca en el punto de venta.
10. Hace más eficiente el uso de la flota de transporte.



(Fig.1.4-3) Distribución

- 11.** Racionalización de envases y embalajes => 100% de la superficie del palet.
- 12.** Simplificación en el manejo de los inventarios.
- 13.** Disminución de los costos de manipulación, almacenamiento y transporte.
- 14.** Mayor rentabilidad por metro cuadrado de almacenamiento.
- 15.** Optimización en general de la logística de almacenamiento y distribución.





(Fig.1.4-4) Optimización en la logística de trasnportes.

- 16.** Permite desarrollar programas de reabastecimiento continuo.
- 17.** Estrecha relaciones entre clientes y proveedores a lo largo de la cadena.

1.4.2 Unidades de carga y aspectos constructivos del PALET.

Según la norma UNE ISO 445 la definición de palet es la siguiente: **"plataforma horizontal rígida, cuya altura está reducida al mínimo compatible con su manejo mediante carretillas elevadoras, transpaletas o cualquier otro mecanismo elevador adecuado, utilizado como base para**

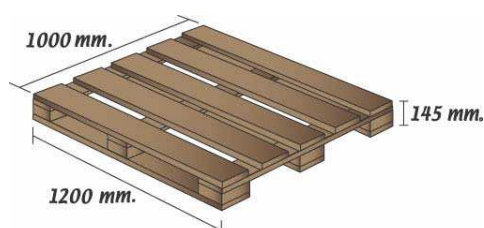
	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

agrupar, apilar, almacenar, manipular y transportar mercancías y cargas en general".



Se emplearán para "**Piensos O´Couto**" las unidades base de carga con las Dimensiones máximas (800*1200) mm para el mosaico de 3 sacos por capa, y de (1000*1200) para los mosaicos de 4 y 5 sacos por capa.

Se considera como base transporte la siguiente Unidad de carga:

- **EUROPALET:** De madera de dimensiones máximas estándar y con una altura mínima de 145mm. Tiene cuatro entradas libres (dos a cada uno de los lados estrechos), y el hueco de cada entrada, tiene una altura mínima libre de 100mm, incluso con carga máxima.



(Fig.1.4-5) Palet Europalet

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	


1.5 KUKA SYSTEM.

Es imposible imaginarse la industria sin robots que realicen el paletizado. Entre todos los tipos y marcas existentes, los robots de paletizado son capaces de realizar cualquier tipo de tarea, ya sea durante el embalaje o el transporte de productos de panadería, al apilar pesadas cajas de bebida o durante la manipulación de tetrapaks en la periferia de instalaciones de llenado.



[Kuka Robots Ibérica](#), es una empresa perteneciente a la corporación internacional **Grupo Kuka**, está especializada en el desarrollo de la ingeniería para el producto final y la producción de sistemas para su realización, además de la formación de los trabajadores y soporte para el mantenimiento y modernización de las líneas de producción. Dentro de su gama de paletización para el sector del envasado y la alimentación, la compañía cuenta con una gran oferta en soluciones de manipulado, paletizado y final de línea, que ofrecen “gran productividad y reducción de costes”.

Los robots trabajan casi de forma interrumpida siendo las paradas de producción y las averías prácticamente inexistentes. Así, se reducen los gastos y la fluctuación de personal. Los buenos resultados conseguidos desde siempre en la producción son también el futuro en la logística. Los sistemas con robot efectúan tareas complejas como el embalaje, el paletizado, o la preparación de envíos, que hasta ahora sólo se podían lograr con mucha mano de obra o con instalaciones de técnica compleja, con mucha más rapidez y eficacia de manera totalmente automatizada durante las 24 horas del día. Y todo ello con un grado de flexibilidad único más allá del paletizado; pues, a diferencia de las máquinas e instalaciones

ROBITEK.System Dpto.Robótica	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

especializadas en un producto, los robots de kuka se pueden adaptar continuamente a diferentes versiones de productos o a productos nuevos.



(Fig.1.2-1) Kuka System.

La tecnología innovadora de máxima precisión, ha revolucionado la producción industrial. No sólo porque han automatizado muchos procesos gracias a su unidad de control basada en PC, sino porque ante todo son capaces, con su estructura modular, de adaptarse a exigencias particulares. Su reducido peso ahorra energía y permite alcanzar una velocidad de trabajo y una aceleración máximas, incluso con cargas elevadas, logrando tiempos de ciclo más breves. El peso de diversos modelos se ha visto reducido todavía más gracias a innovadoras técnicas constructivas

Adaptado a casi cualquier tipo de herramienta, se sujetan a la muñeca del robot mediante una brida estándar certificada según normativa que permite cambiar la herramienta con rapidez y capaces de manejar un sinfín de productos y embalajes.

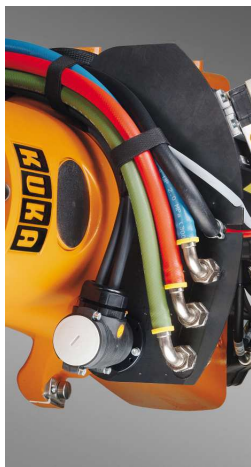
(Fig.1.2-2) Muñeca Kuka.

En los robots de paletizado de cuatro ejes, el brazo principal y el brazo paralelo son de plástico reforzado con fibra de carbono (prc) especialmente ligero y robusto. En estos modelos, la brida de la muñeca se encuentra paralela al suelo, haciendo que sea óptima para aplicaciones de paletizado.



La unidad de control es de manejo sencillo, siendo compatible con la tecnología de Windows.



Su instalación y puesta en servicio resulta siempre fácil y sencilla, siendo una tecnología abierta incorporada mediante Windows, posibilitando las conexiones en RED y el diagnóstico por vía telemantenimiento. Disponen por defecto de la tecnología DEVICENET para las conexiones con el entorno.



(Fig.1.2-3) Paquete de energía.

Kuka, ofrece los paquetes de energía adaptados a casi el 90% de todo tipo de instalaciones. Los paquetes de energía estándar son válidos para la mayoría de las instalaciones, pues disponen de conexión para alimentación neumática, hidráulica, gas,



	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

agua y conexión eléctrica mediante Harting. Todo ello sin interferir en los movimientos y con posibilidad de ser sustituido rápidamente, ya que incorporan bridas de sujeción que protegen en todo momento las conexiones.



(Fig.1.2-4) Bridas de sujeción.

1.5.1 El kr 180-2 pa

Empleado para cargas pesadas, estos robots son ligeros, rápidos y fuertes. Los robots de paletizado **kuka kr 180-2 PA** han sido concebidos para realizar tareas a alta velocidad y con elevadas cargas. Su cinemática de óptimo diseño ha sido confeccionada para tareas de paletizado. El robot de 4 ejes destaca por el mínimo mantenimiento y servicio que precisa. Todos los sectores se benefician de su elevado valor añadido y tiempo de amortización. Los **kr 180-2 PA** son capaces de apilar hasta 3.000 mm en europalets. Las características principales son las siguientes:



(Fig.1.2-5) KR 180-2 PA.

Capacidad de carga:	180 kg.
Carga Adicional:	50 kg.
Carga adic. brazo / brazo oscilación / colum. giratoria :	250 kg.
Máx. Alcance:	3.200 mm.
Repetibilidad:	± 0,25 mm.
Número de ejes:	4.
Unidad de control:	KR C2.
Peso (sin unidad de control), aprox:	1.200 kg.
Posición de montaje:	Piso.

1.5.2 Campo de Aplicaciones (Paletización).



- **PALETIZADO DE CAJAS.**



(Fig.1.2-6) Paletizado de cajas

En Nestlé Nespresso en Orbe, Suiza, se fabrican cápsulas Nespresso. Dado que la empresa con la inauguración de esta fábrica aumentó su capacidad de producción en cinco veces, era necesario integrar una técnica de paletizado eficiente.

El paletizado de las cajas fue asumido por tres robots KUKA del tipo KR 180 PA. Estos cogen cuatro cajas al mismo tiempo y las apilan sobre europalets de acuerdo con la figura de apilado correspondiente.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

- **PALETIZADO DE ATADOS DE BEBIDA.**

En la industria de la bebida deben paletizarse grandes cantidades de distintos atados. Por este motivo la cervecería Carlsberg en Suecia se decidió por un robot KUKA del tipo KR 500. Este campeón de peso pesado, que dispone de una capacidad de carga de 500 kg, apila allí los atados con botellas de plástico. La alta capacidad de carga es necesaria porque el robot paletiza siempre una capa completa.



(Fig.1.2-7) Paletizado de bebidas.

- **PALETIZADO DE SACOS.**



(Fig.1.2-8) Paletizado de sacos.

La empresa BASF buscaba una solución para paletizado automático para los concentrados de vitaminas A, que se utilizan preferentemente para alimentación de animales en sacos de 25Kg. También para alimentos y productos farmacéuticos.

El centro de la automatización lo conforma un robot de seis ejes de KUKA del tipo KR 125, equipado con una garra de horquilla para sacos. El criterio fundamental para la selección del sistema automático fue el poco espacio disponible.



- **APILADO DE EMBALAJES.**

La empresa CD Cartondruck AG, un líder en la fabricación de embalajes para artículos de cosmética, buscó una solución automatizada. Respecto a los requerimientos, se pedía, sobre todo, tiempos de ciclos cortos, flexibilidad y fácil manejo.

El trabajo es efectuado por dos robots KUKA de paletizado tipo KR 180 PA. Cada robot carga palets sobre dos transportadores a rodillos. Los KR 180 PA colocan separadores intermedios de cartón entre las capas, para garantizar siempre la estabilidad óptima.



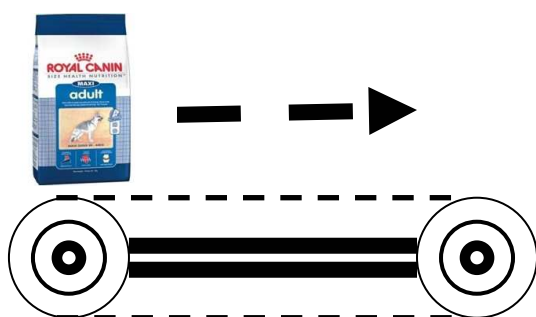
(Fig.1.2-9) Apilado de Embalajes.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

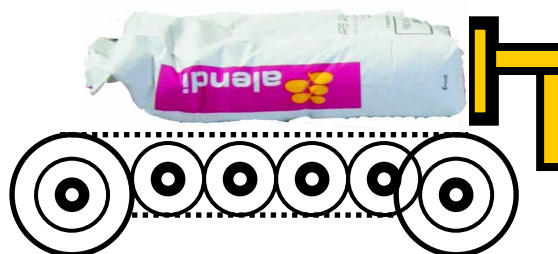
2 SINOPTICO DE LA INSTALACIÓN.

Se presentará en este apartado a modo descriptivo, una pequeña memoria de la instalación y su funcionamiento, para posteriormente y con más detalle presentar todos los elementos integrantes de la instalación de Robot Paletizado.

Para **"Piensos O´Couto"** podemos observar la distribución de los componentes principales en los esquemas sinópticos que se adjuntan más abajo. La entrada de sacos viene a través de la **"Cinta transportadora del cliente"** (**Fig.2-1**), ensacados con el producto en el interior y de pie. Estos son transportados hasta la máquina denominada **"Vuelca sacos"** que activará un accionamiento neumático mediante final de carrera, tumbando así el saco para que entre preparado a los rodillos de alimentación (**Fig.2-2**).



(Fig.2-1) Cinta transportadora Cliente



(Fig.2-2) Rodillos de Alimentación después del vuelcasacos.

El saco debe entrar a la mesa de alimentación (**Fig.2-3**) con la parte inferior del mismo haciendo tope con el soporte final del transportador, de manera que todos los sacos entren en la misma posición y sea el robot quién realice el giro para ir conformando el dibujo del palet, dichos rodillos de entrada están preparados y adaptados a las características geométricas de la garra del robot (**Fig.2-4**).



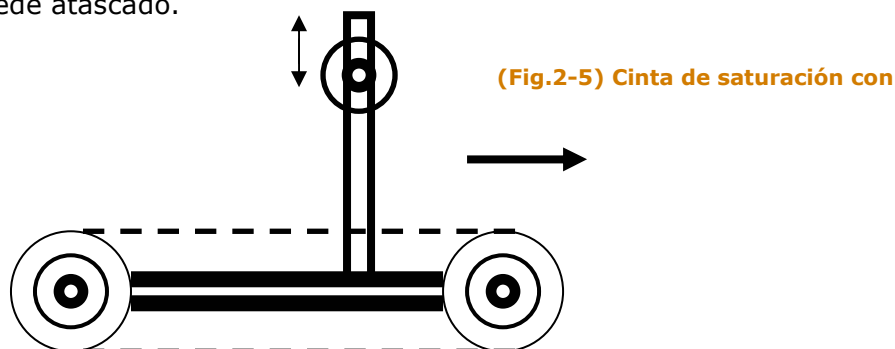
(Fig.2-3) Mesa alimentadora de sacos



(Fig.2-4) Garra del robot

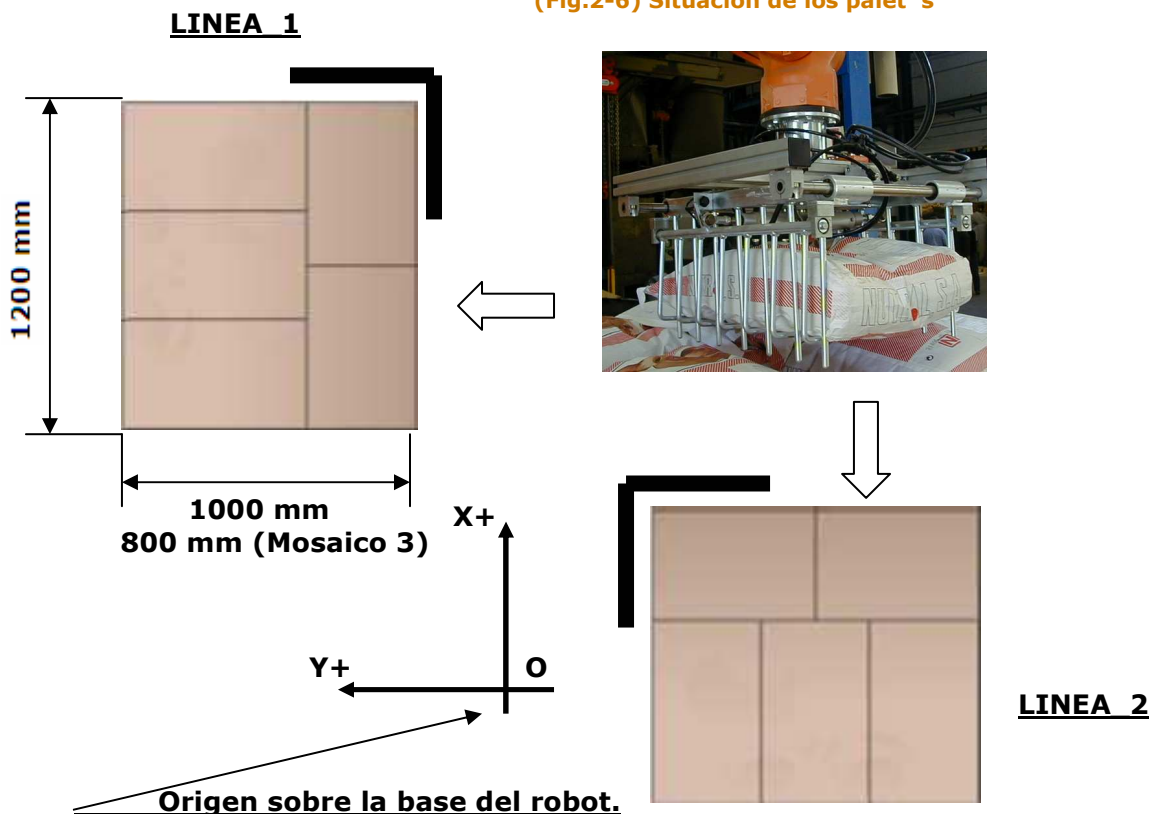
DEDOS DEL ROBOT



Antes de la entrada del saco a la mesa alimentadora de rodillos, se dispone de otra mesa de saturación (Fig.2.-5) en donde un nuevo saco de entrada esperaría en el caso de que el robot no se hubiera llevado aún el saco en los rodillos de alimentación. Esta cinta a su vez dispone de un rodillo prensa neumático que es accionado en el momento que se está desplazando el saco para distribuir el producto y que entre a los rodillos alimentadores lo más homogéneo posible; esto es importante para conformar un buen dibujo del palet. El Rodillo prensa es neumático compuesto por un cilindro; al cual, mediante regulación de caudal se le ajusta la presión que ha de ejercer sobre los sacos para que el rodillo actúe sin que el saco se quede atascado.



Una vez producida la entrada del saco a los rodillos de alimentación, este será recogido por la garra del robot y transportado al palet correspondiente. Los palet's tal como se muestran en la figura (Fig.2-6), se encuentran posicionados en el suelo, siendo el carretillero quién recoja el palet manualmente una vez termine el paletizado en el mismo. Ambos palet's se posicionan en el suelo con ayuda de unas escuadras

(Fig.2-6) Situación de los palet's

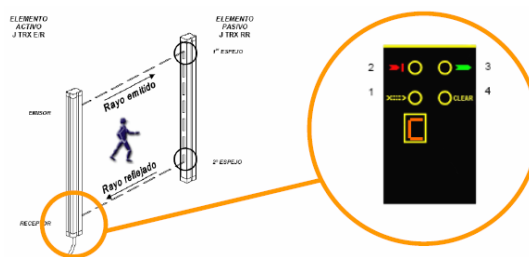
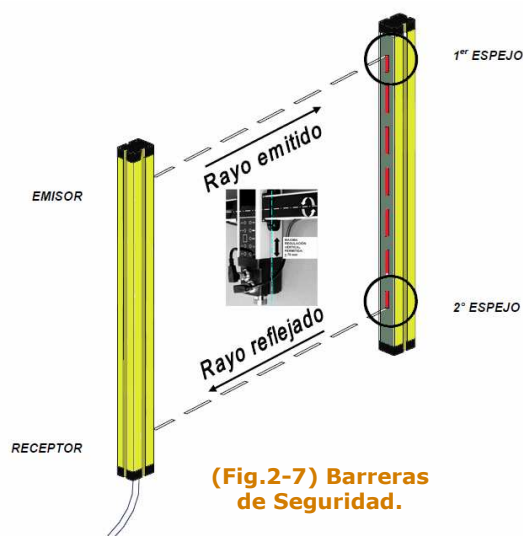


	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

que facilitan que la posición sea siempre la misma, esto es vital para que los sacos depositados por el robot se queden en el palet siempre en la misma posición, así mismo habrán de emplearse siempre el mismo tipo de palet o europalet, siendo las medidas de este de 1200 * 1000 (mm) para los mosaicos de 4 y 5 sacos o 1200*800 (mm) para el mosaico de 3 sacos.

El Robot; representado en el centro principal de la instalación, se encuentra ubicado lo más paralelo y perpendicular posible a los palet's donde se ha de paletizar, esta posición estratégica es vital pues nos permitirá en un futuro poder modificar los puntos de dejada de saco teniendo en cuenta única y exclusivamente las coordenadas en X e Y. No obstante; y tal como veremos más adelante, se crearon bases de coordenadas X e Y paralelos a los lados del palet para facilitar dicha tarea.

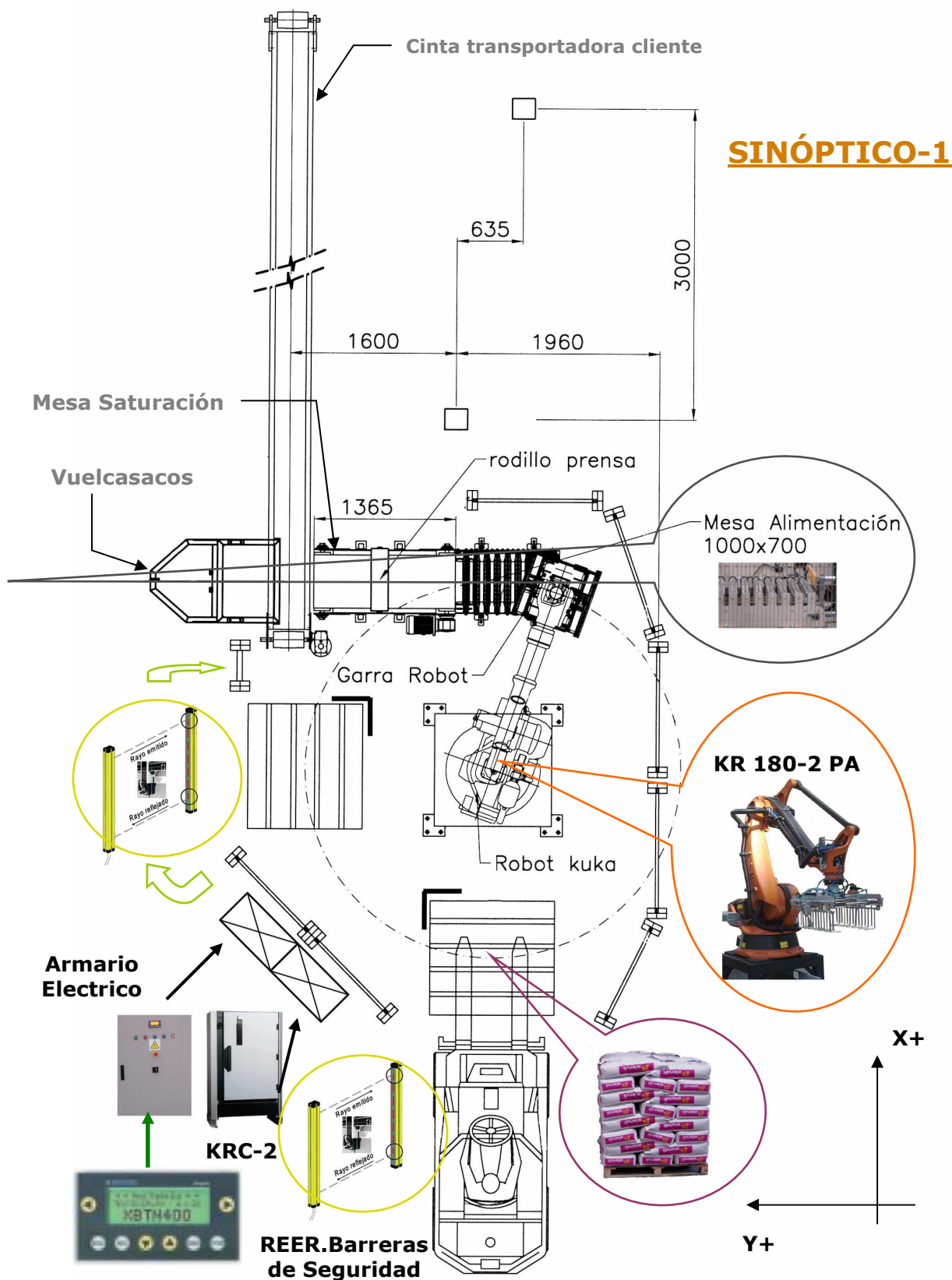
El recinto se encuentra totalmente vallado; según mandan las normas para estos tipos de instalaciones robotizadas. En este caso no hay puertas de acceso, siendo la única forma de entrada a través de las barreras de seguridad que se encuentran en ambas entradas de recogida de palet's. Estas barreras se compone de





dos elementos verticales (receptor y emisor) o barrera principal, cualquier obstáculo entre ellos hace saltar las seguridades parando el robot, necesitando estar libre de obstáculos y rearmadas para volver a poner en marcha la instalación nuevamente. Existe la posibilidad de bloquear el paletizado en ambas líneas, de esta manera el robot se encontrará fuera

de la zona bloqueada, pudiendo acceder al recinto sin peligro alguno y bajo la responsabilidad del operario.

Como últimos elementos principales, quedan los dos armarios que conforman la instalación. El armario eléctrico del robot denominado **"KRC2"** y el armario eléctrico de control que incluye el PLC y la pantalla XBT (**Ver. Sinóptico_1**). Ambos se encuentran adjuntos en una parte del vallado, de manera que faciliten el manejo de la instalación y puesta en servicio de las diferentes opciones desde un punto visual óptimo. Desde el KRC2, seleccionaremos el programa de paletizado para el control automático mediante PLC, mientras que en el armario eléctrico encontraremos los mandos principales de marcha y paro, así como una pantalla desde la que podremos manejar todas las opciones de paletizado, cambiar el tipo de mosaico, el número de capas por palet, etc. Ambos elementos serán descritos con todas sus opciones en los sucesivos capítulos.

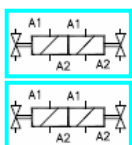


	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

En el siguiente esquema adjunto (**Ver.Sinóptico_2**) se representa la distribución de lo que sería la parte de sensores de la periferia. Existen sensores de proximidad (Fig.2-8), tanto palpadoras (detectan por distancia de aproximación) como reflex (detectan cuando se cortan el haz sobre un espejo). Otro tipo de sensores empleados en la periferia son los magnéticos empleados en la detección de la posición de los elementos de la garra y un detector o presostato situado a la entrada del robot (Fig.2-9) que indica cuando el nivel de presión en la garra es la adecuada para ser usada.

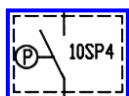


(Fig.2-8) Sensor de proximidad.



Representa las dos electroválvulas de memoria de las que se componen la mordaza del robot. Esta posee dos cilindros neumáticos para la apertura y cierre de sus dos componentes principales que serían los dedos y las manos tal como se detalla en la descripción de este elemento hardware.

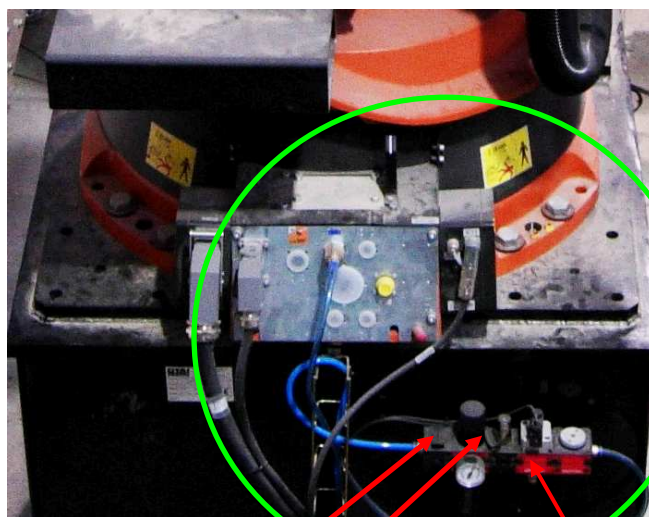
▼ Junto a los cilindros de apertura y cierre de la mordaza, se acoplan dos sensores (**8**) y (**9**) de tipo magnético posicionados de tal manera que nos indiquen mediante señal, la posición de reposo de estos dos elementos; es decir, "**Dedos abiertos**" y "**Manos abiertas**" pues será necesaria asegurar dicha posición cada vez que el robot vaya a por un nuevo saco.



▼ El siguiente elemento, representaría en los esquemas eléctricos adjuntos, el detector de presión (**1**) de aire en la mordaza del robot. Esta no tendrá que bajar de 2 bares de presión para que los elementos de la mordaza funcionen correctamente. Por lo tanto en este rango tendrá que ser ajustado.

(Fig.2-9) Entrada presión del robot.



Vemos en la imagen adjunta, donde estaría situada la entrada de conexión neumática con la unidad de mantenimiento formada por el regulador de caudal y el sensor de presión. La entrada de aire se produce mediante una electro válvula que se activará mientras haya tensión en el armario eléctrico; y así proporcionar presión de aire a los elementos de la mordaza del robot. A la izquierda se encontraría el Harting de conexiones eléctricas que alimentan las bobinas y detectores de la garra.



Regulador Presión.

Presostato.

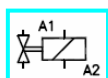
Electroválvula.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	



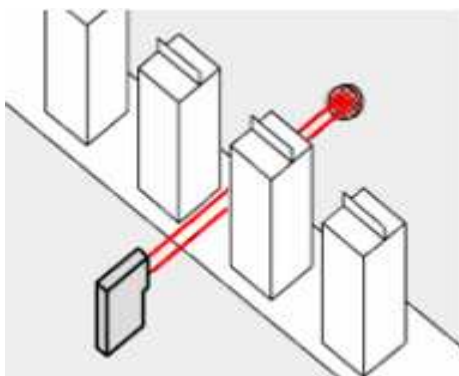
La electroválvula de entrada de aire, no depende de las seguridades de la instalación, puesto que la mayor precaución es evitar que la mordaza del robot se quede sin aire para evitar la caída del saco que se encuentra en ese momento sujeto por el robot.

Para **"Pienso O' Couto"** a nivel neumático nos encontramos con el cilindro y la electroválvula de simple efecto que hace actuar el rodillo prensa.

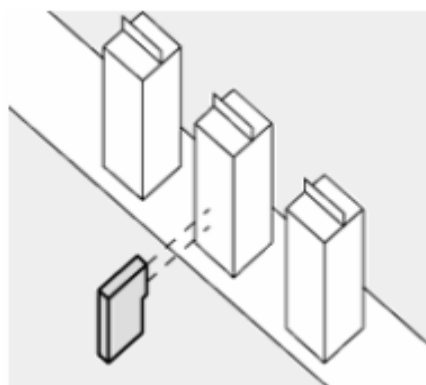


Este elemento actuará cuando el detector de presencia **(6)** o fotocélula palpadora indique que el saco se encuentra en posición. Al ser una electroválvula de simple efecto, la señal a la bobina tendrá que ser mantenida mientras tengamos presencia en **(6)** o determinado también por el programa del PLC.

(4), (5) Esta fotocélulas de presencia son de tipo reflex, esto significa que nos darán presencia cuando el saco obstaculice un haz no visible que refleja sobre un espejo situado al otro lado (Fig.2-10 y 2-11). El usar este tipo de fotocélulas, aseguran que la presencia se ha producido por algún obstáculo sin haber falsas detecciones, esto es muy importante cuando se trabajan con distintos materiales y diferentes colores, ya que una fotocélula reflex detecta algo que es opaco, y no depende del color como pudiera pasar con las fotocélulas palpadoras normales. Su lógica de funcionamiento es negada; es decir, nos dan señal **"true"** si se produce una reflexión sobre el espejo, y no señal o **"false"** si hay algún obstáculo.





(Fig.2-10) Reflex en Reposo 'true'



(Fig.2-11) Reflex actuando 'false'

Mediante la señal **(4)** y determinado por el tiempo de filtrado a través del PLC, paramos el saco en la posición de recogida por el robot, asegurando que dicha posición haga tope sobre el soporte, tal como se mostró anteriormente.

Con la señal **(5)** se asegura un pequeño acumulo en la línea; es decir, si la fotocélula **(4)** se encuentra libre de obstáculo la fotocélula **(5)** no actuará, entrando el saco siempre hasta la posición de recogida, pero si la presencia en **(4)** indica que el saco aún no ha sido recogido, la presencia en **(5)** actuará como un segundo paro para un segundo saco, permaneciendo en esta posición hasta que la mesa de alimentación de rodillos vuelva a estar libre.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

▼ **(7)** Esta señal actúa como paro de la cinta transportadora del cliente, en caso de que la mesa de saturación y la mesa de alimentación estén ocupadas por algún objeto, de esta manera si hay presencia en ambas mesas, en el momento que algún saco sea detectado por la presencia en **(7)** se parará la cinta del cliente, evitando así que entren nuevos sacos, ya que tanto la mesa de saturación como la mesa de alimentación se encuentran paradas. Bajo estas circunstancias existen las siguientes posibilidades:

1. La cinta del cliente siempre estará en marcha (controlada por los operarios) siempre y cuando alguna de las presencias en **(4)** o **(5)** este libre, ya que esto significa que puede entrar algún saco y no hay motivo para realizar un paro de línea.
2. En caso de que la presencia en **(4)** esté ocupada, la cinta del cliente se parará con la presencia en **(7)** para evitar que un nuevo saco entre en la mesa de saturación que se parará con la presencia en **(5)** (**Ver Paro en línea del capítulo 6.3.3 -> Salidas digitales**).
3. La mesa de saturación estará funcionando siempre que la instalación este en marcha y cualquiera de las dos presencias **(4)** ó **(5)** se encuentre libre. Solamente se parará si el operario realiza un paro de la instalación o ambas presencias **(4)** y **(5)** se encuentran ocupadas.
4. Por último la mesa de alimentación de rodillos se pondrá en marcha si lo está la instalación, no hay presencia en **(4)**; puesto que si no, no estaría libre, y se ha producido una presencia anterior en **(5)**. Una vez producida la marcha solamente se detendrá después de un tiempo con la presencia en **(4)**.

Ni que decir tiene, que tanto la mesa de alimentación de rodillos como la cinta de saturación, dependerán en todo momento de sus protecciones térmicas que también provocarían un defecto saltando alarma en la instalación. También ambas mesas pueden moverse mediante desplazamientos en manual.

▼ **(2), (3)** Estas fotocélulas determinan la existencia de carga en los palet's, de esta manera cada vez que el robot finalice un palet; y tras su retirada y colocación de uno nuevo, será condición indispensable y necesaria no detectar presencia antes de iniciar el paletizado en los nuevos palet's. Actúan por lo tanto como elementos de seguridad cada vez que iniciemos la ejecución de una nueva paletización.

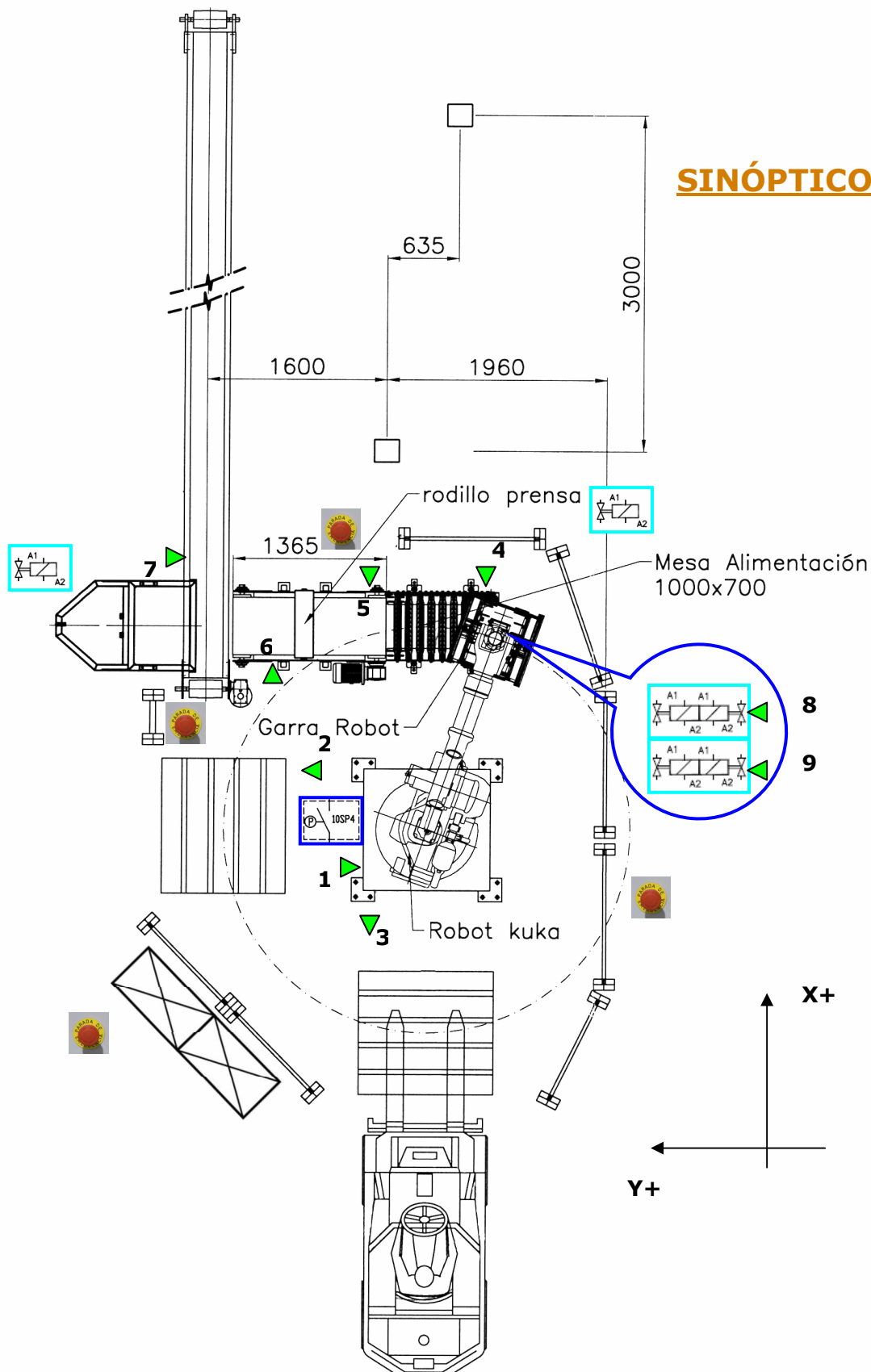


Los últimos elementos representados se denominan **"Setas de emergencia"** (**Fig.2-12**), estos dispositivos; como se puede observar, se encuentran distribuidos por toda la periferia, armario eléctrico general y vallado hasta un total de cuatro setas de emergencia, ya que son elementos importantes de seguridad.

(Fig.2-12)
Seta de
emergenci

Cualquier accionamiento sobre uno de ellos desactiva las seguridades provocando un paro inmediato en toda la instalación, a excepción de la unidad de mantenimiento que en ningún momento debe quedarse sin aire para evitar la caída del saco que este en la garra.

SINÓPTICO-2



3 COMPONENTES HARDWARE (Paletizado).

Se presentará en esta sección una descripción detallada de los elementos y partes fundamentales de la instalación que nos llevará a una fácil comprensión sobre la funcionalidad de la misma. Después de este capítulo estaremos en disposición de abordar la parte de programación en sí, ya que conoceremos en detalle la funcionalidad de los elementos sobre las que se compone la instalación.

Dada su vital importancia como elemento en cuanto al papel que juega en la arquitectura del robot paletizador, comenzaremos detallando la herramienta, garra o Mordaza que se encuentra acoplada en la brida del eje **A6** del robot, y sobre la cual actuaremos en todo momento para la recogida y dejada de sacos.

3.1 HERRAMIENTA DEL ROBOT (Garra).

La mordaza del robot y sus accionamientos nos permiten mediante la apertura y cierre de dos cilindros posicionar el saco en su interior para posteriormente cogerlo y volverlo a depositar en el palet correspondiente. Esto se consigue mediante los siguientes elementos integrantes en la misma:

- **Cilindros neumáticos.** Dos cilindros neumáticos son gobernados por dos electroválvulas de memoria con dos posiciones. Mediante estos cilindros; representados en la figura (Fig.3-1 y 3-2), somos capaces de realizar el desplazamiento de las partes móviles de la herramienta que nos permite actuar sobre los sacos. Estos son denominados de la siguiente forma.



1. Apertura/Cierre DEDOS. Los dedos del robot (Fig.3-1) son los elementos móviles en forma de "L" y son los que se sitúan por debajo de los rodillos de la mesa de alimentación para que el peso del saco se apoye sobre ellos en el momento de la recogida y desplazamiento. Realizan por lo tanto, un movimiento de apertura y cierre como si de los dedos de la mano se tratara. La mordaza posee hasta siete dedos en forma de "L" a ambos lados, por lo tanto los rodillos de alimentación tendrán que poseer un mínimo de siete huecos, de manera que los dedos del robot queden por debajo de la superficie de apoyo del saco en el momento de la recogida. En la siguiente figuramos simulamos la parte trasera de un saco que ha llegado a su posición y que se encuentra sobre los rodillos de la mesa. Vemos que la entrada del robot a la recogida ha de hacerlo con los dedos abiertos para posteriormente cerrarlos.



(Fig.3-1) Desplazamiento de los dedos.



(Fig.3-2) Desplazamiento de las manos.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

2. Apertura/Cierre MANOS. Las manos (Fig.3-2) son los otros elementos móviles de la mordaza, y tienen forma de varilla recta. Este cilindro tiene la misión de sujetar el saco una vez que los dedos lo sostienen en el momento de la recogida y la dejada en el palet. A diferencia de los dedos, estos se encuentran ligeramente por encima de manera que no puedan golpearse con los rodillos en el momento de la recogida. Observemos en la figura la acción que ejercen las manos en el momento de recogida del saco, en este caso sería semejante a sostener algo con el puño cerrado.

En la figura anterior, podemos divisar como en la posición **"Dedos Cerrados"** y **"Manos Cerradas"** el saco ya se encuentra perfectamente ubicado en el centro de la mordaza, de manera que el primer movimiento que realiza el robot hacia arriba, el saco se queda apoyado sobre los dedos y no se moverá al estar sujeto por las manos.

Sobre el punto de recogida determinamos que una vez que el robot esta situado justo encima del saco, los dos cilindros actuaran conjuntamente de manera que se cierran a la vez para ganar tiempo de ciclo. Mediante programa esperaremos el tiempo suficiente para asegurar el cierre total de la garra antes de empezar a subir (Fig.3-3); esto es debido a que no disponemos de sensores magnéticos que nos indique la posición cerrado de ambos cilindros.

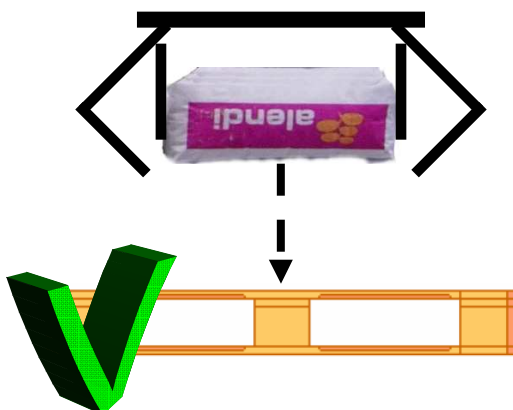


(Fig.3-3) Posición Manos y Dedos Cerrados.

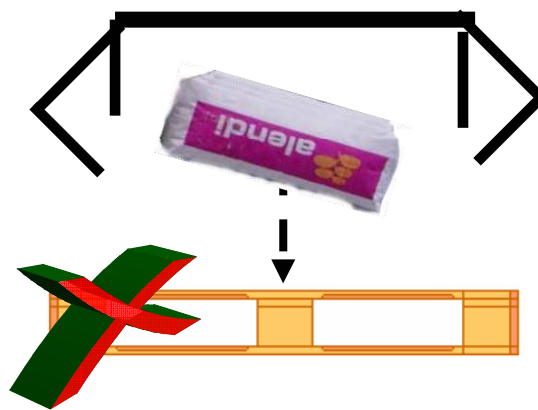


(Fig.3-4) Posición Manos y Dedos Abiertos.



Durante el trayecto ambos elementos permanecerán cerrados, y en el punto de dejada sobre el palet el único elemento que se abrirá serán los dedos, de manera que el saco caerá por su propio peso a la vez que las manos cerradas mantienen el saco equilibrado durante la caída (Fig.3-5 y 3-6).



(Fig.3-5) Dedos abiertos y Manos cerradas.

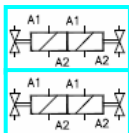


(Fig.3-6) Dedos abiertos y Manos abiertas.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	



- **Electroválvulas.** Integran las bobinas electromagnéticas que gobiernan la apertura y cierre de los dos cilindros. De esta manera disponemos de dos electroválvulas de memoria con dos bobinas cada una. Estas señales se encuentran gobernadas por el robot, por lo que así están referenciadas en anexos del bornero en los esquemas eléctricos, y en el programa siendo estas las siguientes:

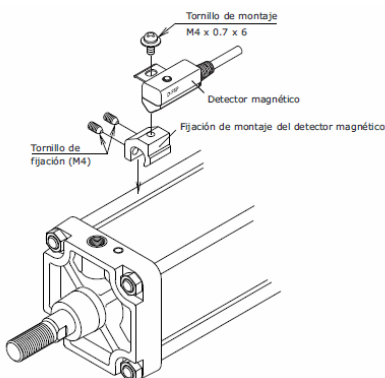


- DO3-1 -> Abrir_Dedos -> \$OUT[19] en el programa del robot.**
- DO3-1 -> Cerrar_Dedos -> \$OUT[20] en el programa del robot.**
- DO3-1 -> Abrir_Manos -> \$OUT[17] en el programa del robot.**
- DO3-1 -> Cerrar_Manos -> \$OUT[18] en el programa del robot.**

En la arquitectura software del robot se detalla todo sobre la síntesis y procesos de ejecución del robot. Por programa se activará una señal de **+24V** a la salida correspondiente para que le llegue tensión a la electroválvula en concreto que queremos excitar.

- **Detectores Magnéticos.** Su utilidad aquí es determinar en todo momento la posición de reposo de la mordaza; esto es, "**Dedos abiertos**" y "**Manos abiertas**" de tal manera que dispongamos de estas señales de seguridad. Mediante programa esta situación es siempre verificada una vez que el robot ha terminado de depositar un saco en el palet, encontrándose por lo tanto en el punto HOME o de espera, y así asegurar que en el momento de recibir permiso para recoger un nuevo saco, la mordaza se encuentra totalmente abierta para entrar en los rodillos de alimentación. De no producirse dicha situación por falta de apertura en alguno de los cilindros, el robot no irá a recoger un nuevo saco y activará al cabo de un tiempo (4 Seg.) una señal de avería que notificará al PLC el estado de no reposo en la mordaza.

- DO2-4 -> Defecto_Mordaza -> \$OUT[12] en el programa del robot -> %I5.11 -> Di_defecto_mordaza -> entrada directa al autómata.**
- DI4-3 -> Manos_abiertas -> \$IN[27] en el programa del robot.**
- DI4-4 -> Dedos_abiertos -> \$IN[28] en el programa del robot.**



(Fig.3-7) Acoplamiento del sensor magnético.



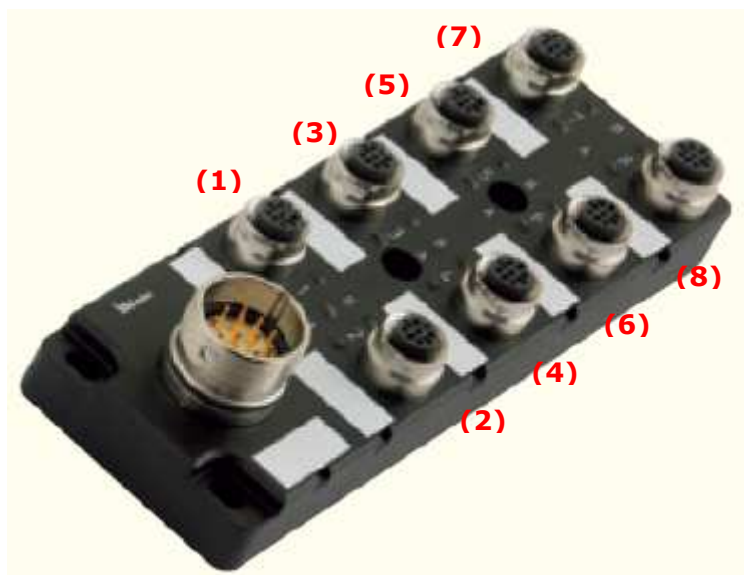
(Fig.3-8) Posición del sensor magnético.

De producirse dicho defecto, se notificaría al operario mediante mensaje en la pantalla Magelis del armario eléctrico, y bastaría con rearmar mediante el botón del armario para tal efecto, dicha avería para que el robot prosiguiera con su ejecución habitual. En caso de rearmar la avería sin haber alcanzado la posición de reposo de la mordaza, el defecto volvería a saltar nuevamente transcurridos 4 Seg.

En las figuras (Fig.3-7 y 3-8), se muestra en detalle como van posicionados dichos detectores en los cilindros. Al ser magnéticos lo que se detecta es la posición del vástago, de esta manera acoplándolo al final o principio del recorrido del mismo se detecta una posición u otra.

- **LUMBERG. Conexiones eléctricas.**

Los lumberg (Fig.3-9), se denominan a las base de conectores mediante los cuales se cablean las entradas y salidas. Todo lo que nos hace falta es una base a través de cual conectar las diferentes señales que empleamos en la garra, y otro conector al cual se le cableará mediante una manguera todos los hilos necesarios que llevaremos a través del paquete de energía hacia la base del robot, donde podremos acceder fácilmente mediante algún bornero o Harting que llegará al armario eléctrico o KRC2. Estas señales serán cuatro salidas para la apertura y cierre de los dos cilindros y dos de entrada para la detección de la posición de reposo de la mordaza, a parte está la alimentación para la cual necesitaremos otros dos hilos y un tercero para el cable de masa. Para esto nos valdría con preparar una manguera de 8 hilos aunque siempre es recomendable dejar hilos de reserva.



SALIDAS.

- (1). Abrir_Manos.
- (2). Cerrar_Manos.
- (3). Abrir_Dedos.
- (4). Cerrar_Dedos.

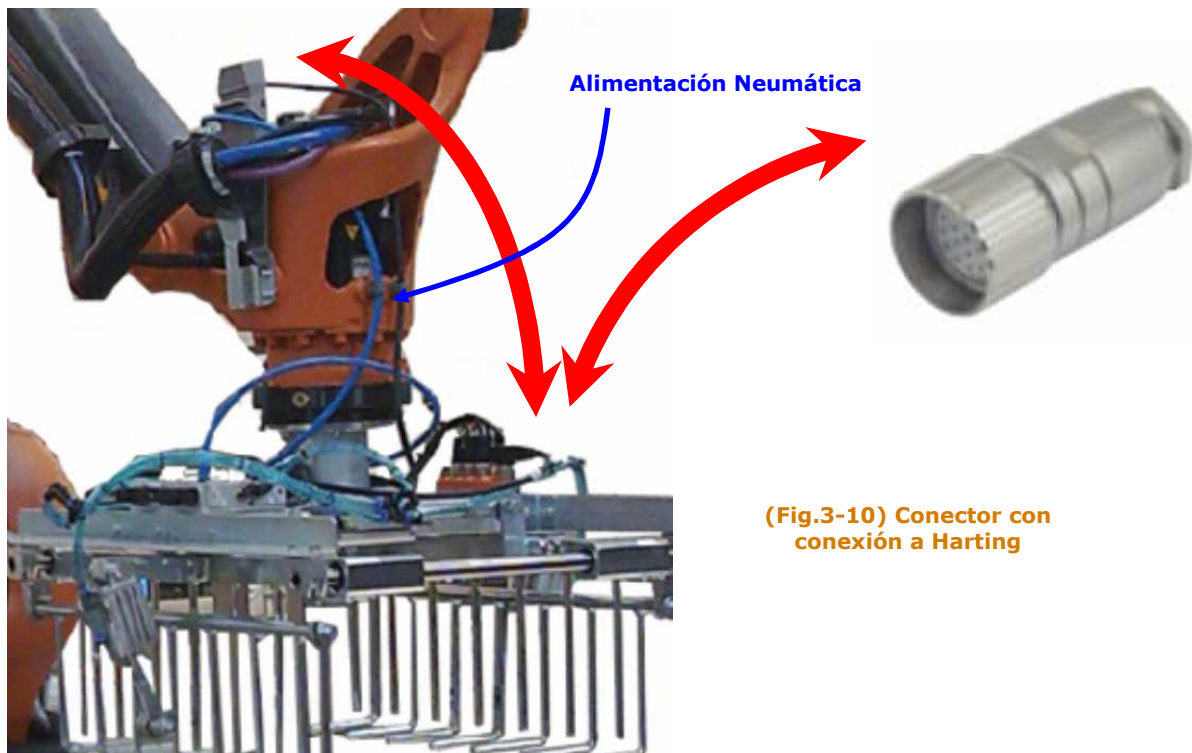
ENTRADAS.

- (5). Manos_abiertas.
- (6). Dedos_abiertos

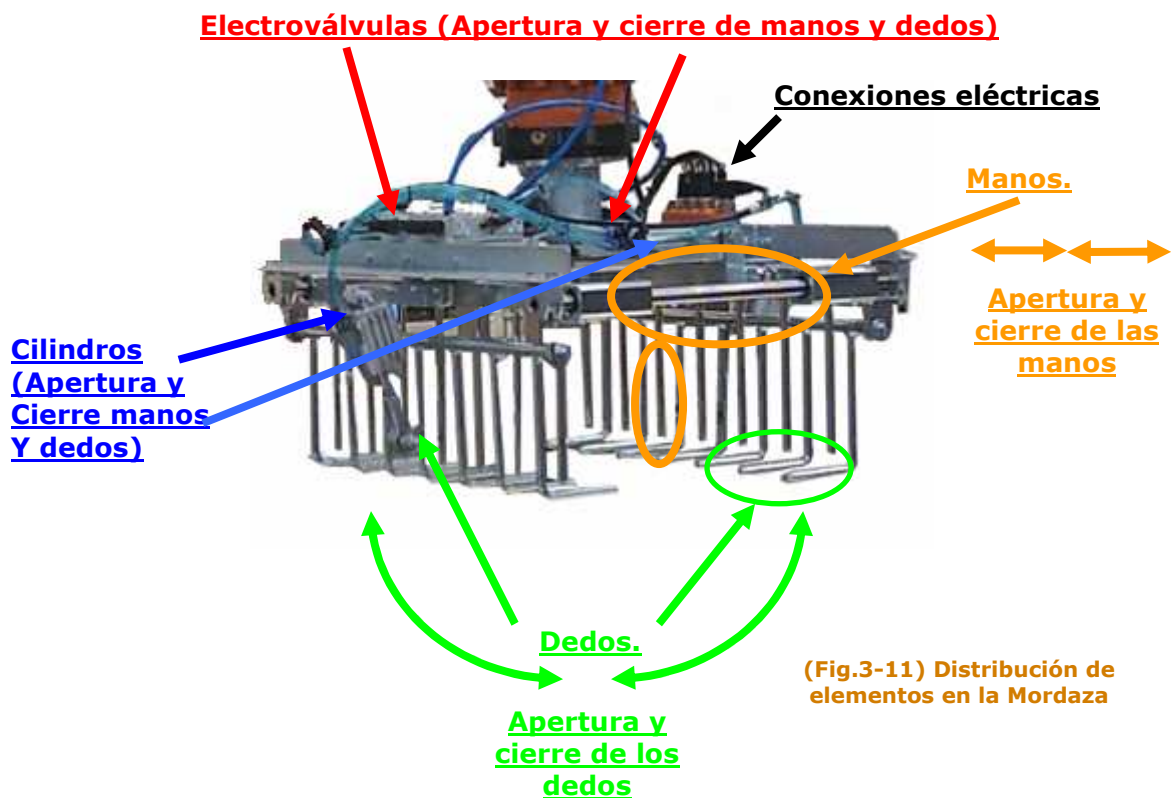
En cada una de estas posiciones, se acoplarán conectores que unirán la base con los sensores o bobinas electromagnéticas.

(Fig.3-9) LUMBERG de conexiones eléctricas.

El conector hembra irá al Harting o conexión del paquete de alimentación (Fig.3-10) que llegará hasta la base del robot, donde podremos hacer las conexiones con los armarios cómodamente.

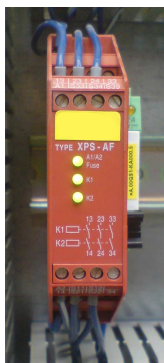


Finalmente y a modo de resumen se mostrará en la siguiente figura (Fig.3-11) la distribución de los elementos importante de la mordaza del robot.



3.2 SEGURIDADES GENERALES (Barreras protectoras).

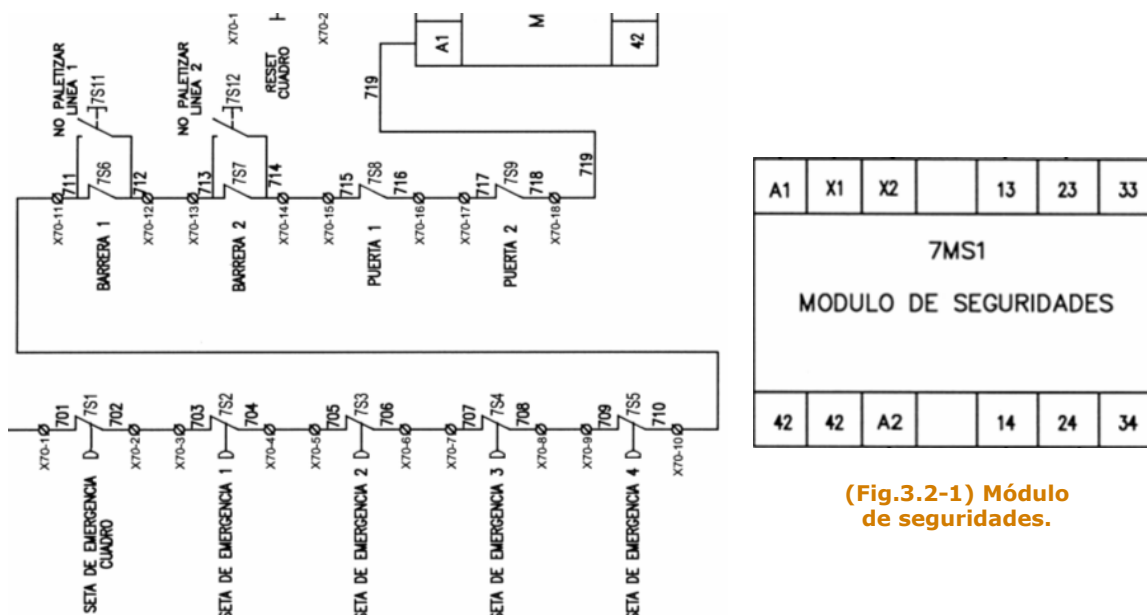
Una parte importante en toda instalación que incluya un robot, son el tema de las seguridades. Analizaremos en este punto aquellos puntos importantes sobre las seguridades que dispone la instalación, desde el paro del robot hasta las barreras de seguridad que nos permiten el acceso al recinto. Para ello haremos uso de las seguridades incluidas en los planos eléctricos donde identificaremos fácilmente los elementos integrantes.



(Fig.3.2-1) Módulo de seguridades.

Si observamos los esquemas adjuntos, el elemento principal y desde donde se gobiernan las seguridades de la instalación, es el módulo de seguridades (Fig.3.2-1) integrado en el armario eléctrico. Desde este módulo se activan los contactores de potencia que suministran energía a los motores de la instalación, y se supervisan de manera constante todas las señales de seguridad, setas de emergencia y barreras. De manera que de detectarse algún fallo, inmediatamente se quitaría potencia en el interior del armario eléctrico, y se abrirían los dos canales de seguridad que llegan al robot. (Ver. Anexos Planos eléctricos).



Entre los terminales **A1-13** del módulo se encuentra cableada la serie de seguridades compuesta por las cuatro setas de emergencia y cada uno de los contactos cerrados de las dos barreras de seguridad. Entre **A1-A2** el módulo debe ser alimentado para poder ser rearmado y activar posteriormente los contactores de potencia y de mando.



(Fig.3.2-1) Módulo de seguridades.

(Fig.3.2-2) Seguridades Eléctricas.

En caso de que pulsemos una seta de emergencia, se abrirá el circuito serie mediante los contactos **SETA DE EMERGENCIA CUADRO, SETA DE**

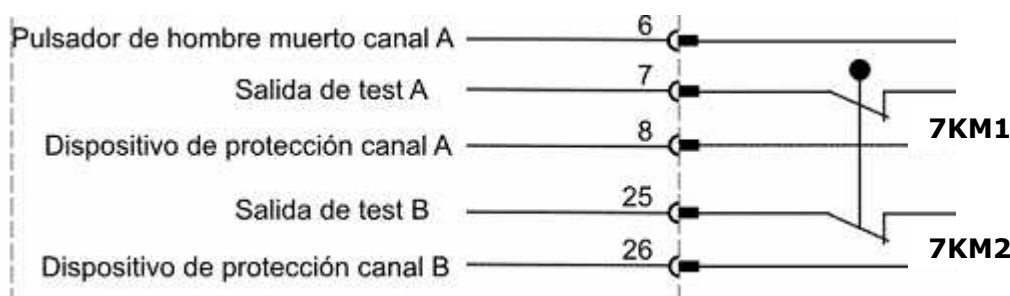
	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

EMERGENCIA 1, SETA DE EMERGENCIA 2 o SETA DE EMERGENCIA 3, de manera que el módulo no podrá ser alimentado entre los terminales **A1-A2**. Si cruzamos por las barreras de seguridad cuando estas se encuentran rearmadas, se abrirán sus contactos cerrados **BARRERA 1** o **BARRERA 2** según corresponda desactivando así las seguridades. Existen en los esquemas una cuarta seta de emergencia y dos contactos pertenecientes a los micros de dos puertas de los vallados, que tendrán que ser puenteados en el bornero del armario, al no disponer de puertas en los cerramientos.

Si el circuito serie está cerrado. Podremos rearmar el módulo mediante el pulsador provisto para ello en el armario **RESET SEGURIDADES** de manera que se produzca un puente entre los terminales **X1-X2**. Si todo está correcto se alimentarán los rele **7KA1** y **7KA2** suministrando energía a los contactores **7KM1 (potencia)** y **7KM2 (Mando)** (Ver **ANEXO Planos Eléctricos**).

3.2.1 Rearme de las seguridades del robot.

En el momento que **7KM1** y **7KM2** se alimentan, a través de dos contactos de rele, se cierran los canales de seguridad del bornero del robot (**CONECTOR X11- del capítulo sistema KRC-2**).





(Fig.3.2-3) Canales de seguridad del bornero X11.

El cierre de estos dos rele puenteará las señales **7-8** y **25-26** del conector **X11** (Ver punto 4.2.1.5 Bornero-X11) del robot, y ha de ser simultáneo en ambos canales, ya que de producirse en un canal antes que en el otro el robot no rearmaría y tendríamos que volver a desactivar el módulo de seguridades para volverlo a rearmar. De esta manera y después de haber rearmado la periferia, procederemos a rearmar el robot mediante el botón "**Reset de Alarmas**" que es una entrada directa al PLC (**%I1.3**) tal como veremos en el siguiente capítulo, donde el robot volverá a estar en disposición de ejecutar movimientos.

Generalmente serán necesarias dos pulsaciones para rearmar el robot. La primera de ellas activará las seguridades internas del armario **KRC-2** y entrarán los accionamientos. Una vez que las seguridades del robot han entrado será necesario volver a producir un segundo rearme para la confirmación de los mensajes de alarma. (Ver **Capítulo Automático Externo**).

3.2.2 Rearme de las barreras de seguridad.

Hemos hablado hasta ahora de cómo rearmar el módulo de seguridades, pero estos elementos o barreras de seguridad también precisan ser rearmados para que los contactos cerrados **BARRERA 1** y **BARRERA 2** se cierren y nos permitan

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

rearmar el módulo. Podemos definir ya un proceso sencillo a la hora de rearmar las seguridades de la instalación de manera ordenada:

1. Rearmar Barreras de Seguridad.

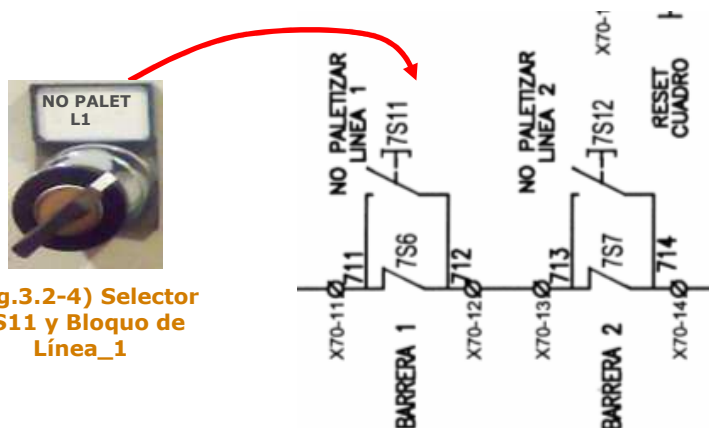
Con las barreras, OK

2. Rearmar módulo de seguridad.

Con el módulo, OK

3. Rearmar armario KRC-2.

El como rearmar las barreras de seguridad será lo que nos ocupe a continuación. No obstante; y analizando nuevamente los contactos cerrados de las barreras de seguridad, vemos que estos se encuentran en paralelo con los accionamientos **7S11 (Fig.3.3-4)** y **7S12** correspondientes a las señales del PLC **%I1.13(DI_Robot_no_Paletiza_L1)** y **%I1.14 (DI_Robot_no_Paletiza_L2)**. Estas señales físicas, cableadas mediante selectores de mando, se utilizan cada vez que queremos bloquear una línea de manera que el robot no tendrá permiso para paletizar en esa zona. A parte de bloquear la línea, puentea la seguridad de la barrera, de manera que podamos entrar al recinto sin que salten las seguridades y la otra línea no se vea afectada.



(Fig.3.2-4) Selector 7S11 y Bloqueo de Línea_1

Esta situación se dará cada vez que el operario no quiera trabajar en una zona, o cada vez que el robot haya finalizado un palet, donde el operario tendrá que desactivar la barrera correspondiente, ya que de no ser así, las seguridades saltarían y pararíamos la otra línea de paletizado. La forma de desactivar dicha barrera o de bloquear una línea es mediante el contacto en paralelo **7S11 (línea 1)** o **7S12 (línea 2)** que activamos con el selector.



Hay que tener en cuenta que para quitar un palet; haya terminado o no de paletizar, será siempre preciso desactivar la barrera correspondiente para entrar el recinto, de manera que la otra línea no se vea afectada. El hecho de desactivar la barrera mediante el selector, implica que una vez que coloquemos un palet nuevo, antes de volver a colocar el selector en su posición tendremos que rearmar la barrera de la forma que indicamos más abajo, por que de no ser así, las seguridades saltaran. Después de rearmada la barrera podremos volver a poner el selector en su posición, y daremos al botón "Palet nuevo" que se encuentra en el armario eléctrico para que el robot comience a paletizar nuevamente en la zona.

El estado de las barreras de seguridad es accesible mediante unos visores **led** dispuesto en ambos lados, emisor y receptor para la evaluación de las anomalías. Si algún elemento interrumpe el camino óptico generado entre ambos,

el receptor desactivará sus salidas, provocando la caída del módulo de seguridades. Su colocación y montaje tendrá que ser de manera que queden simétricas en cuanto a la altura y la inclinación, ya que de no estar correctamente montadas nunca se activarían. La separación máxima permitida entre el receptor y emisor será de unos 2,5 metros.

- **Señales en el emisor.**

En el momento de la activación, durante 5 seg. Todos los Leds se encenderán para a continuación mostrar su estado actual de funcionamiento.




LED	SIGNIFICADO FUNCIONAMIENTO NORMAL (LED ENCENDIDO)	COLOR
	Barrera de Seguridad en test	AMARILLO
	Barrera de Seguridad en funcionamiento	VERDE
	Fallo detectado	ROJO
	Largo alcance seleccionado *	NARANJA

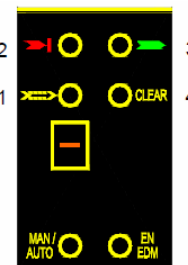
(Fig.3.2-5) Estados del emisor

- **Señales en el Receptor.**

En el momento de la activación el visualizador mostrará el número **8** permaneciendo todos los leds encendidos, para después mostrar la configuración seleccionada. A continuación se muestra cual sería el estado normal de visualización.

En caso de anomalía quedaría encendido sólo el led rojo, con el indicador visualizando una "C" intermitente o un "F" indicando la anomalía.

JANUS "J"	Weak (1)		Situación de Señal débil recibida	Señal recibida OK	NARANJA
	Break (2)		Barrera ocupada, salidas OSSD en OFF	-	ROJO
	Guard (3)		Barrera libre, salidas OSSD en ON	-	VERDE
	Clear (4)	CLEAR	Barrera libre, salidas OSSD en OFF (el receptor está en espera de una señal de Reanudación)	-	AMARILLO

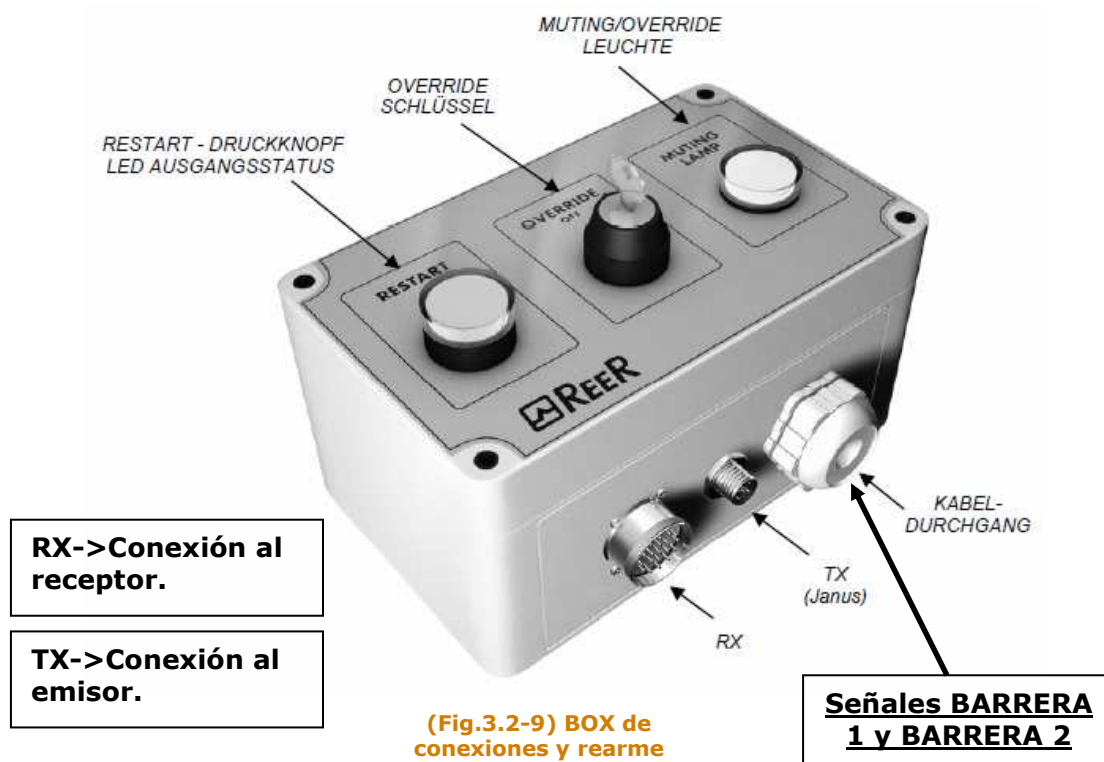
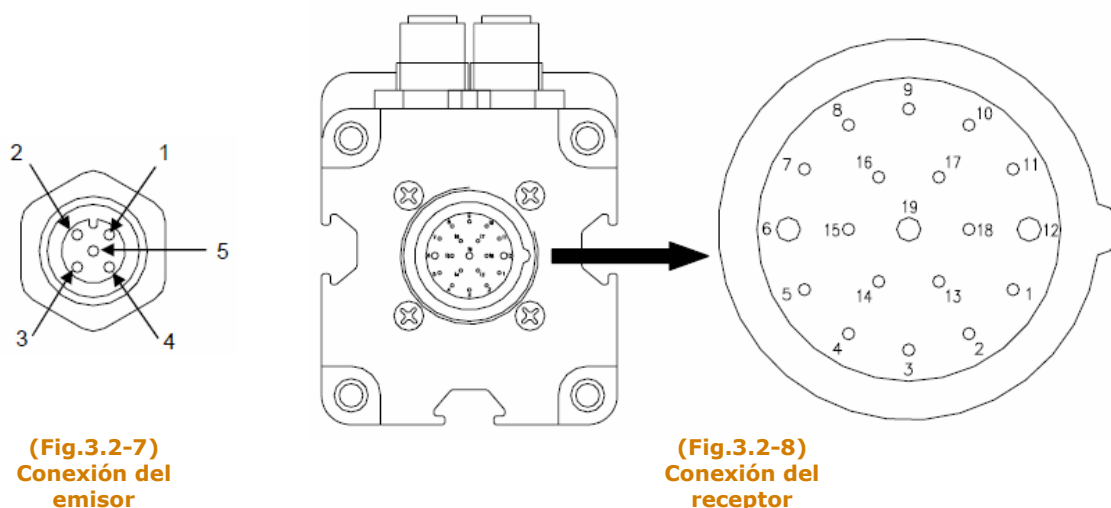


(Fig.3.2-6) Estados del receptor

La denominación **OSSD** hace referencia a las salidas de seguridad de las barreras.

ROBITEK.System Dpto.Robótica	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

Por último las conexiones de los diferentes elementos de la barrera se realizan a través de los denominado **BOX**. En cuyo interior se encuentra la tarjeta electrónica con las señales **BARRERA 1** y **BARRERA 2** que tendremos que llevar al armario eléctrico previo cableado, para conectar el emisor con el receptor mediante los cables y lumberg proporcionados por el fabricante. Para la conexión del emisor se empleará el conector de 5 pines proporcionado por el fabricante (Fig.3.2-7), y para la conexión del receptor el conector hembra de 19 pines (Fig.3.2-8).



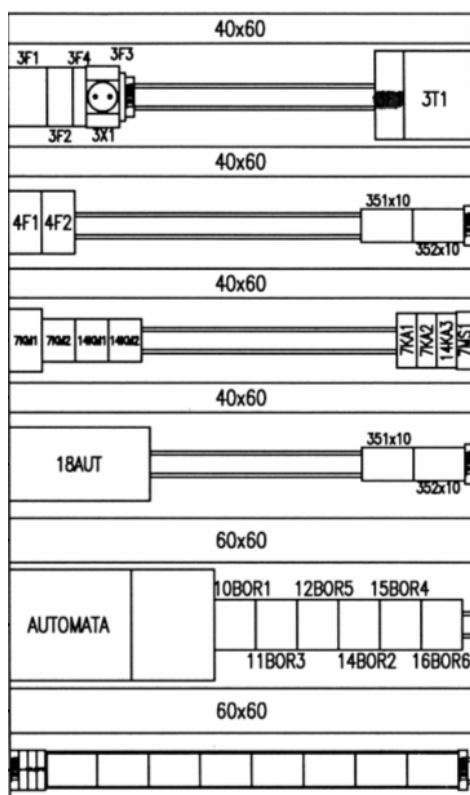
La manera de rearmar la barrera de seguridad, una vez que no hay obstáculo, consistirá en mantener accionado el botón de **RESTART** (Fig.3.2-9) de manera que el piloto amarillo del pulsador; que se encuentra parpadeando (defecto), se quede fijo indicando el estado OK de la barrera y cerrando su contacto o salida de seguridad correspondiente.

3.3 ARMARIO ELECTRICO. (Magelis XBT-N401)

El armario eléctrico de la periferia compone el elemento principal de manejo y mando de la instalación, ya que desde ella tendremos acceso y controlaremos todos los elementos integrantes (incluido el robot), a través de los accionamientos previstos y del terminal de dialogo en el cual se detalla en los sucesivos apartados.

Dentro de un armario de estas características, encontramos todos los elementos y mecanismos eléctricos para el cableado de señales y alimentación de todos y cada uno de los componentes integrantes. En los esquemas eléctricos adjuntos (**Ver ANEXO: Esquemas Eléctricos**), se detalla principalmente la distribución de estos.



3.3.1 Descripción de componentes.



(Fig.3.3-1) Distribución de componentes.

El interior (Fig.3.3-1) se compone de canalizaciones y regletas donde se van integrando todos los componentes. Para "**Benjamín Valledor**" estos serán:

- **Fuente de Alimentación (24V) (3V1).** Está será la fuente de suministro de energía para todos los elementos y sensores que se alimenten a 24 V con corriente continua. Estará por lo tanto dimensionada acorde con el consumo total necesario siendo este **24V, 5Amp**. Con ella alimentaremos el autómata o PLC, así como la Interface XBT para la comunicación con el usuario.
- **Trasformador (3T1).** Alimentado de dos de las tres fases de la acometida trifásica, este elemento transformador de energía suministra la tensión de entrada necesaria a la fuente de alimentación para la transformación de corriente. Sus características son: **380v/220v (Primario/Secundario) 315VA**.
- **Enchufe 220V (3X1).** Destinado a la conexión de aparatos eléctricos externos a la instalación como por ejemplo los portátiles necesarios la puesta en marcha o algún tipo de luminaria.
- **Fusible (3F3).** Se utiliza para las protecciones del enchufe auxiliar siendo de "**1Amp**" de esta manera el transformador estará protegido en caso de consumo excesivo en dicho terminal.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

- **Magneto térmico (3F4) unipolar.** Empleado a la salida de la fuente de alimentación. Permite cortar o suministrar de corriente continua la instalación. Su valor es de **"6Amp"** por encima del máximo permitido por la fuente, ya que no actúa como protección si no como conmutador
- **Magneto térmico (3F2).** Conectado a la salida del transformador de tensión, habilita la entrada de la fuente de alimentación. De esta manera podemos probar potencia y utilizar el enchufe sin habilitar ningún elemento de control alimentado a 24v. Su valor es de **"6Amp"**
- **Magneto térmico (3F1).** Este magneto térmico está destinado para hacer las funciones de interruptor general pues está cableado a la salida del seccionador principal del armario, permitiendo por lo tanto la entrada de alimentación trifásica al armario. Su valor es de **(20Amp)**.
- **Contactores Principales (7KM1 y 7KM2).** Dependen totalmente del módulo de seguridad de manera que el contactor de potencia **7KM1** conectado después del magneto térmico **3F1**, permite la distribución de la acometida trifásica al resto de elementos de potencia como son los motores. Mientras que el contactor de mando **7KM2** habilita mediante software la entrada de seguridades que le llega al autómatas.

(Fig.3.3-2)
Protección
térmica.



Mediante un contacto abierto del **7KM1** y otro del **7KM2** se habilitan las seguridades del robot para que pueda ser rearmado siempre y cuando lo este el módulo de seguridades del armario de control.

- **Protecciones térmicas (4F1 y 4F2).** Estos elementos se sitúan aguas abajo del contactor **7KM1** para la entrada de potencia a los motores de la instalación. Son ajustables de tal manera que bajo un consumo excesivo de los motores, saltarán provocando el corte inmediato de energía eléctrica. Y el paro de la instalación (Fig.3.3-2).

- **Contactores de Potencia (14KM1 y 14KM2).** Conectados tras las protecciones térmicas son los elementos de mando a partir de los cuales activando la salida correspondiente del autómatas pondrán en marcha los motores de la cinta y rodillos de saturación llevando la potencia a ellos (Fig.3.3-3).

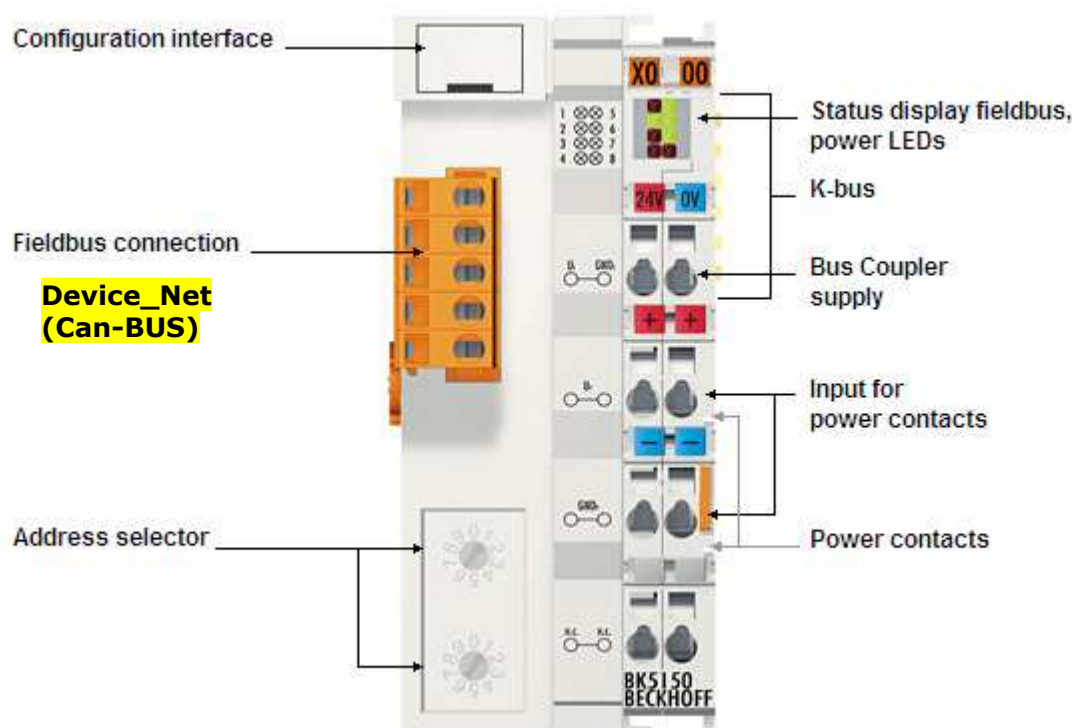


(Fig.3.3-3)
Contactor de
Potencia.

- **Módulo de Seguridades (7MS1).** Es el elemento principal en la cadena de seguridades. Mediante un control permanente de las señales de seguridad directas (setas de emergencia y barreras de seguridad), desactivan la cadena de mando quitan potencia a la instalación en caso de producirse una alarma provocada por el accionamiento de una seta de emergencia o por la activación de una barrera de seguridad en modo de operación normal.
- **Reles de Protección (7KA1 y 7KA2).** Son los reles con contactos libre de potencial, que permanecerán activados mientras lo esté el módulo de seguridad. Mediante contactos abierto permiten la activación de los contactores principales **7KM1** y **7KM2**.
- **AUTOMATA TSX371020 DTK (Fig.3.4-7).** Componente fundamental que integra todo el software de gestión de la instalación, y también el control y manejo del robot de manera automática. Será por tanto el elemento el cual

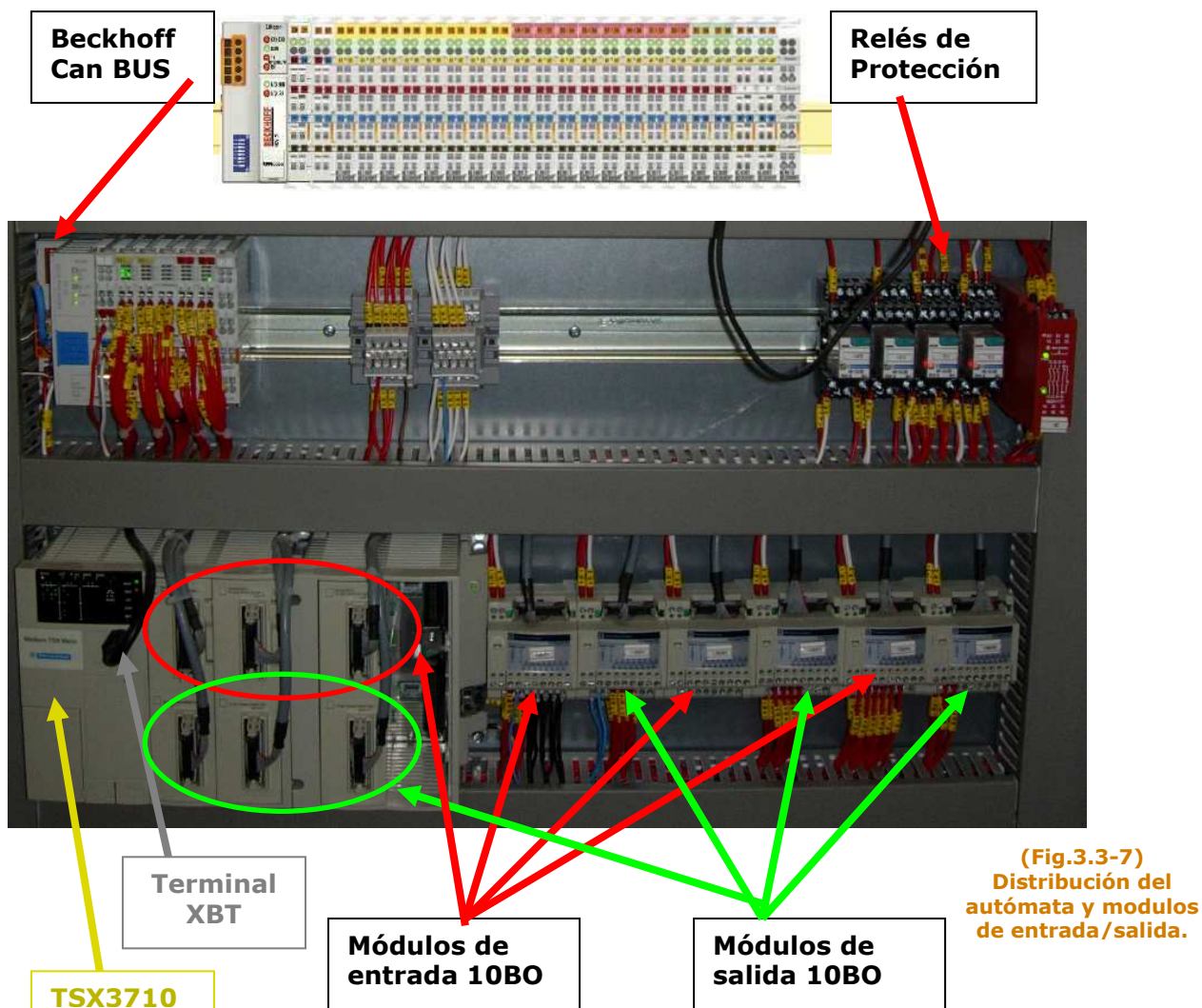
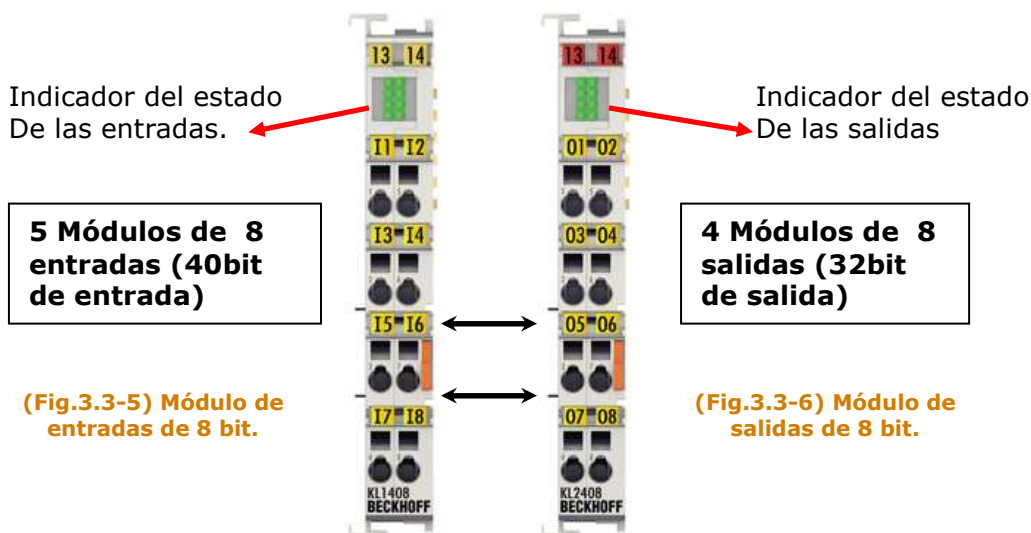
a través de la señales de estado que le llega por sus entradas, determinará las salidas de mando en función de lo que haya que hacer en todo momento.

- **Bornes de entrada 10BOR1/R3/R5 (Fig.3.4-7).** Estos tres borneros tipo Telefax se conecta cada uno a los módulos de entrada del autómata tipo **TSX DMZ28DTK**, siendo estos módulos de 16 entradas de 24V / 12 salidas de 0,5 Amp. En dicho módulo irán cableadas cada una de las entradas o sensores de la periferia con entrada directa al autómata, y también todas las señales de entrada de comunicación con el robot con salida directa de este. (Ver ANEXO: Programa de PLC).
- **Bornes de salida 10BOR2/R4/R6 (Fig.3.4-7).** Estos tres borneros tipo Telefax se conecta cada uno a los módulos de salida del autómata tipo **TSX DMZ28DTK**, siendo estos módulos de 16 entradas de 24V / 12 salidas de 0,5 Amp. En dicho módulo irán cableadas cada una de las salidas (reles, bobinas de contactor, electroválvulas) de la periferia con salida directa del autómata, y también todas las señales salidas del PLC con entrada de control y comunicación con el robot. (Ver ANEXO: Programa de PLC).
- **Módulo Beckhoff (18AUTO) (Fig.3.3-4).** Se trata del bornero del robot, destinado para la interconexión con la periferia mediante la configuración del archivo "io_sys.ini" (Ver capítulo 4.5.5 Drivers E/S). Se cablearán todas sus entradas y todas sus salidas, para ser transmitidas finalmente por medio del cable de comunicaciones **Can-BUS**. (Ver capítulo 4 SISTEMA ROBOT KUKA).



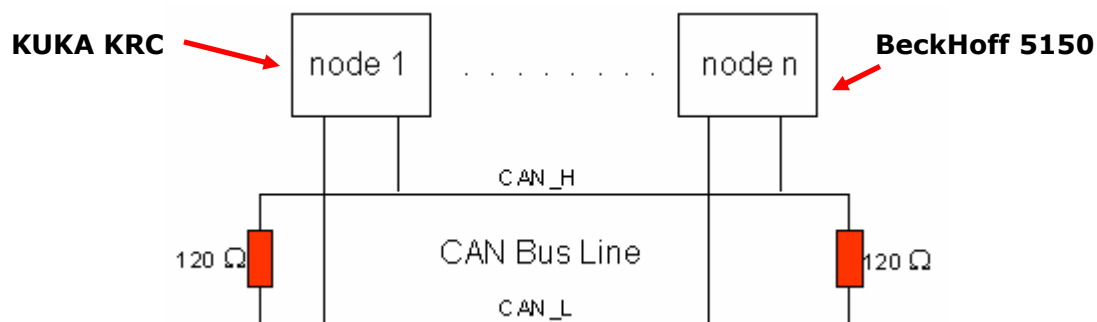
(Fig.3.3-4) Conexiones
Módulo Beckhoff.

El módulo **18AUTO** estará formado y conectado junto con los módulos de entradas y salidas (Fig.3.3-5 y 3.3-6), acoplados fácilmente entre sí. Aunque no todos los módulos son utilizados se sobredimensiona en cuanto a número de entradas y número de salidas para futura expansiones.



3.3.2 Cable CAN-OPEN.

La tecnología de conexiones del CAN BUS integra dos líneas (Canal H y Canal L) del bus conectadas en paralelo con resistencia de terminación de 120Ω.



(Fig.3.3-8) Resistencias de terminación del cable CAN-OPEN.

La velocidad de transmisión estará determinada por la longitud del cable, aunque para nosotros no será ningún parámetro a tener en cuenta, puesto que ambos armarios KRC2 y eléctrico se encuentran contiguos.

El cable se compone de cuatro hilos de conexión con la siguiente distribución (Fig.3.3-9) :

BK51x0 PIN	Función
1	CAN Ground
2	CAN Low
3	Screen(malla)
4	CAN High
5	not used



(Fig.3.3-9) PIN-OUT Conector CAN BUS

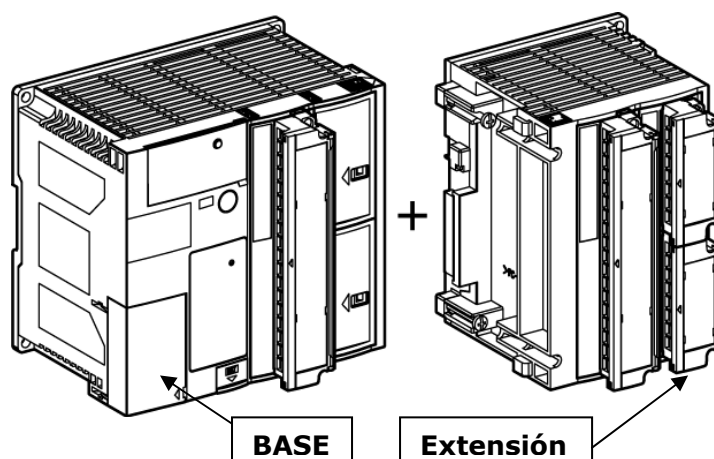
Se muestra junto con la configuración de los hilos, el socket que será empleado tanto en el módulo beckhoff como en el PC del armario KRC-2, y en donde el terminal **Pin 1** será llevado a la masa del Beckhoff. Con este socket tendremos que conectar las resistencias en paralelo de 120Ω entre los terminales **Pin 2 y Pin 4**. O también podíamos usar el cable **ZS1052-3000** de **Beckhoff**.



(Fig.3.3-10) Cable ZS1052-3000

3.3.3 Autómata TSX 3710

El autómata modular **TSX-3710** (Usado en esta instalación), al no incorporar módulos de E/S de serie, permiten una configuración libre al usuario multiplicando sus posibilidades mediante el rack extensible opcional. El TSX37 resulta un autómata versátil para la gran mayoría de aplicaciones, gracias a su amplia gama de módulos de entradas y salidas que maneja.



(Fig.3.3-11) Base y Extensión Autómata TSX.

Los autómatas **TSX 37-10** presentan cinco configuraciones básicas, diferenciadas por el tipo de alimentación así como por el módulo TON implantado en el primer emplazamiento (puede recibir 2 módulos analógicos y 2 módulos de conteo). Estos autómatas pueden recibir un mini-rack de extensión que permite aumentar el número de entradas/salidas locales hasta 192 E/S, e integran un fechador.


En el armario eléctrico de control se encuentran integrados los módulos **28DTKZ** que incorporan 16 entradas digitales de 24V y 12 salidas estáticas de hasta 0,5Amp.

3.3.3.1 Conector terminal RS 485.

El conector terminal RS 485, en formato mini-DIN 8 patillas, permite conectar un terminal del tipo FTX o PC compatible, una impresora o conectar el autómata al bus **UNI-TELWAY**, Modbus. Para ello, ofrece de manera predeterminada la modalidad de comunicación UNI-TELWAY maestro de 9600 baudios.

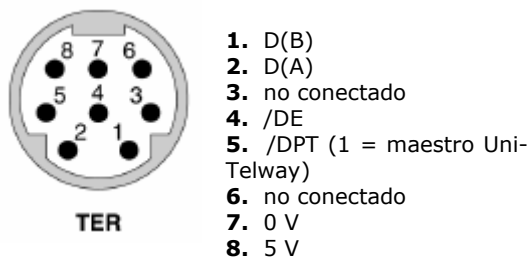
- la modalidad UNI-TELWAY esclavo (Terminal XBT-N401).
- la modalidad caracteres ASCII.
- el protocolo Modbus.

Este conector terminal (Fig.3.3-12) es un enlace RS 485 no aislado formado por un conector mini DIN de 8 puntos. Los autómatas TSX 37-05/08/10 disponen de un conector terminal serigrafiado TER. El conector funciona de forma predeterminada en modalidad UNI-TELWAY maestro y permite conectar 'un

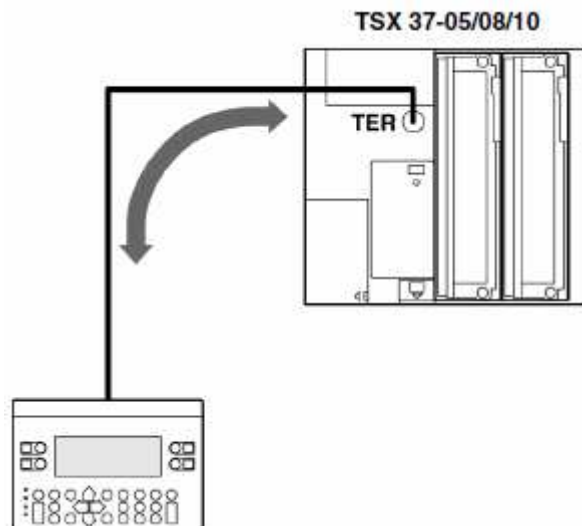
ROBITEK.System Dpto.Robótica	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

terminal de programación y de ajuste a los controladores TSX 37 así como gestionar 'un equipo de diálogo de operador tipo XBT como el analizado más adelante. El paso a la modalidad UNI-TELWAY esclavo, Modbus/JBus o a la modalidad caracteres se efectúa mediante configuración.

Las señales del conector mini-DIN se muestran a continuación:



(Fig.3.3-12) Conector Terminal TSX 37-10.



3.3.3.2 Características del TSX 37-10

Funciones	Número de entradas/salidas TON locales + distantes TSX 07 locales + remotas en el bus AS-i	268 408
	Número de conexiones UNI-TELWAY integradas	1
	Acopladores de comunicación	0
	Fechador	Sí
	Analógica integrada	No
	Contaje integrado - 500 Hz (entrada TON) - 10 kHz	Sí No
Memoria interna	RAM interna que puede guardarse programa (100% Booleano) (1) datos (en RAM interna) constantes	14 Kpalabras 4,7/2,7 Kinst. 1 Kpalabras (2) 128 palabras (2)
	Flash Eprom integrada	16 Kpalabras (3)
Extensión de memoria	Tarjeta PCMCIA	No
Tiempo de ejecución por Kinst.(5)	RAM (100% Booleana)	0,3 ms
	PCMCIA (100% Booleana)	-
Carga general del sistema		1,9 ms
Estructura de la aplicación	Tarea maestra	1
	Tarea rápida	1
	Tarea según sucesos	1 a 8
Bloques de funciones predefinidos	Temporizadores (Timers)	64 (4)
	Contadores	32

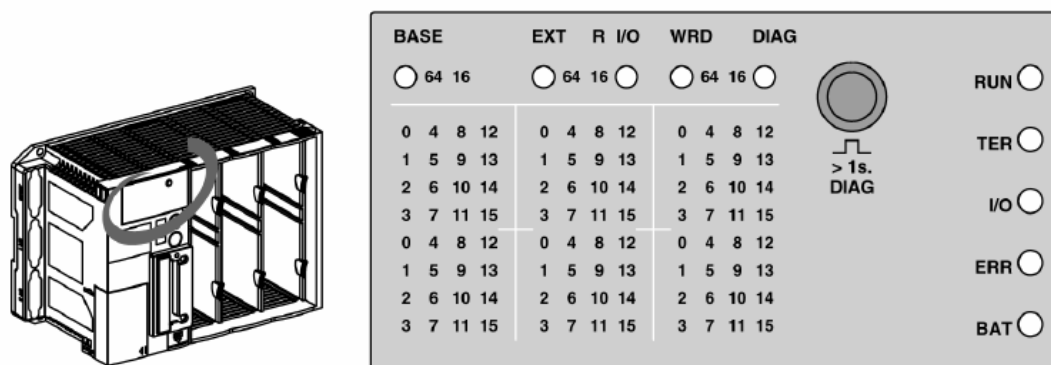
Tabla de características autómatas TSX 37-10.

- (1) El 1er valor corresponde a un programa en List. El 2º valor corresponde a un Programa en lenguaje de contactos.
- (2) Tamaño predefinido, puede ampliarse en detrimento del tamaño del programa De aplicación.
- (3) 15 Kpalabras disponibles para la copia de seguridad de la aplicación + 1 Kpalabras para guardado de los %MW.
- (4) 16 temporizadores como máximo, con la base de tiempo de 10 ms.
- (5) Sin carga general ni gestión de E/S.



3.3.3.3 Bloque de visualización en el TSX 37-10

El bloque de visualización **(3) (Fig.3.3-13)** centraliza todas las informaciones necesarias para el diagnóstico y el mantenimiento del autómatas y de sus módulos. Para ello, se incluyen:

- 8 indicadores de estado que informan del funcionamiento del autómatas (indicadores RUN, TER, I/O, ERR y BAT) y la modalidad de visualización en curso (indicadores R I/O, WRD y DIAG).
- Un bloque de 96 indicadores que permite visualizar:
 1. **En la modalidad de visualización las entradas/salidas locales** (indicador BASE o EXT encendido): el estado de todas las entradas y salidas TON del autómatas y del mini-rack de extensión.
 2. **En la modalidad de visualización de las entradas/salidas remotas** (indicador R I/O encendido): el estado de las entradas/salidas TON de cada esclavo presente en el bus AS-i.
 3. **En la modalidad de diagnóstico** (indicador DIAG encendido): los fallos del "módulo" (parpadeo lento de todos los indicadores asociados al módulo), o los fallos de "canal" (parpadeo rápido del indicador asociado al canal), para las entradas/salidas remotas en el bus AS-i: el estado de cada esclavo (parpadeo de los esclavos en fallo).
 4. **En la modalidad de visualización de objetos** (indicador WRD encendido): el contenido de 16 palabras como máximo %MWi, %SWi o %KWi (estas palabras aparecen en binario o en hexadecimal), el estado de un grupo de 64 bits %Mi, %Si o %Xi, el estado de los bits de entradas y de salidas de los módulos TSX 07 usados como entradas/salidas TON remotas.



(Fig.3.3-13) Bloque de visualización TSX 37

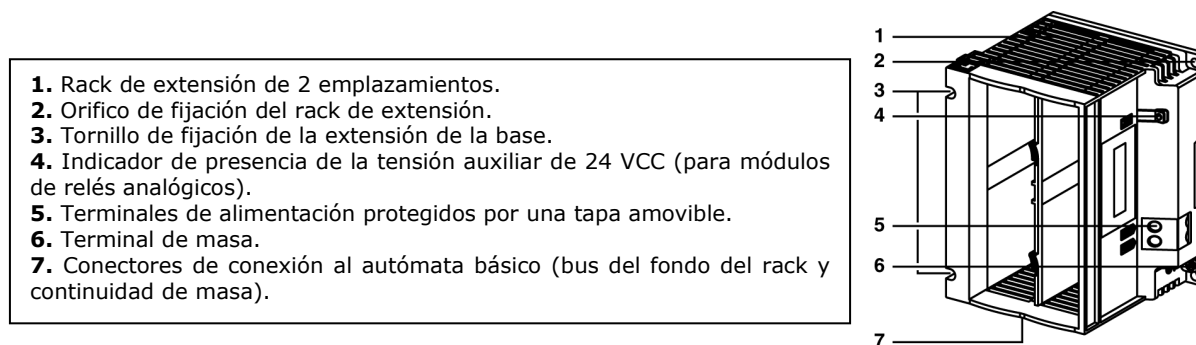
	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

Un pulsador permite visualizar el conjunto de las informaciones y cambiar la modalidad de visualización (Fig.3.3-13).



- **RUN** este indicador (de color verde) se enciende para señalar que el autómatas está en marcha (RUN) y parpadea para indicar que está en la modalidad STOP. Este indicador se apaga cuando no hay ninguna aplicación válida en el autómatas o cuando éste tiene un fallo.
- **TER** Este indicador (de color amarillo) se enciende para señalar que las informaciones se intercambian a través del enlace del terminal. El tráfico por el conector terminal puede dar la impresión de que este indicador parpadea.
- **I/O** Este indicador (de color rojo) se enciende para señalar un fallo relativo a las entradas/salidas:
 1. Fallo de alimentación o disyunción de al menos un canal.
 2. No hay módulo, no está de acuerdo con la configuración o se encuentra fuera de servicio. Para más información acerca de los fallos señalizados por el indicador I/O (fallos de canal o módulo), es necesario pulsar durante más de un segundo el pulsador para pasar a la modalidad de diagnóstico.
- **ERR** Este indicador (de color rojo) se enciende para señalar un "fallo de la CPU" del autómatas. Este indicador parpadea cuando no existe ninguna aplicación válida en el autómatas o cuando hay un "fallo con bloqueo (Véase Detección de fallos a partir de los indicadores de estado del autómatas, p. 226)" del programa de aplicación.
- **BAT** Este indicador (de color rojo) se enciende para señalar que la pila es defectuosa o que no hay pila (opcional). Esta pila que asegura el guardado de la memoria RAM se debe cambiar según un procedimiento específico. Si el bit del sistema %S66 está en estado 1, se inhibe el encendido de este indicador.

3.3.3.4 Mini-rack de extensión

El mini-rack de extensión TSX RKZ 02 (Fig.3.4-15) permite añadir dos emplazamientos a una base de autómatas TSX 37-10/21/22; cada uno de ellos puede recibir un módulo de formato estándar o dos módulos de medio formato.



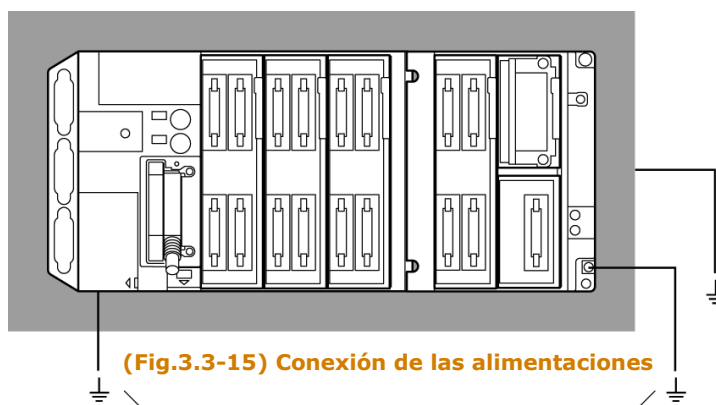
(Fig.3.3-14) Mini-rack de extensión TSX 37.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

3.3.3.5 Conexión de las alimentaciones.

La gama TSX 37 ofrece dos posibilidades de alimentación de los autómatas a través de corriente alterna y a través de corriente continua. Cuando el autómata se alimenta a partir de una red de corriente alterna, ésta no suministra la tensión de 24 VCC al mini-rack de extensión. En este caso, si los módulos de relés o los módulos analógicos están presentes en la extensión, es obligatorio conectar una alimentación auxiliar de 24 VCC en los terminales de la alimentación del mini-rack de extensión (consultar el párrafo "Conexión de las alimentaciones").

La tensión de 24 V suministrada por la base permite alimentar los captadores de ésta y eventualmente a los de la extensión, con la condición que el total del consumo no sea superior a 400 mA. En caso contrario, utilizar igualmente una alimentación de 24 VCC auxiliar. Así mismo todas las partes integrantes deberán estar conectadas a tierra.



3.3.4 Descripción de mando.

En este apartado describiremos los elementos, actuadores y pulsadores integrados en la parte exterior del armario y a partir de la cual, interactuará el operario. La distribución de los elementos es la siguiente:

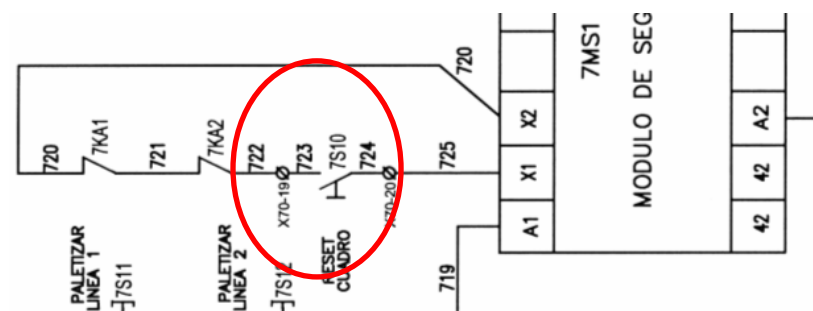


- **Interruptor General.** Mediante este seccionador habilitaremos la entrada de potencia al armario eléctrico. Dicho elemento se sitúa aguas arriba del magneto térmico **3F1** siendo por lo tanto el primer elemento que entra en contacto con la acometida principal a través del bornero.

(Fig.3.3-16) Interruptor General.



- **Pulsador de Rearme (7S10).** Es el pulsador también denominado "**Reset Alarmas**". Después de proporcionar tensión al armario y realizar el rearme de las barreras de seguridad, rearmaremos el módulo de seguridad mediante la pulsación de rearme.



(Fig.3.3-17) Pulsador de rearme y situación en el plano

Dicho accionamiento es una entrada directa al autómata y también se encuentra directamente cableado sobre el módulo de seguridades, de manera que en caso de que este no este rearmado y siempre y cuando no exista ninguna emergencia activa, este pulsador hará un puente entre los terminales **X1, X2** rearmando los rele de seguridad y los contactores de potencia y mando. Una vez que el módulo se encuentre rearmado será necesario volverlo a pulsar para que a través de la Interface "**Automático externo**" el PLC proceda a rearmar el robot, tal como se describió en el capítulo de seguridades, es por lo tanto el pulsador que confirmará los mensajes del robot.

Este pulsador tendrá que ser accionado en caso de rearmar los defectos de la mordaza (**Ver Alarmas XBT**).

%I1.3 -> Reset de Alarmas -> Entrada directa del PLC.



- **Pulsador de Marcha (10S2).** Después de rearmada toda la instalación, se precisará de la confirmación de marcha, a través de este pulsador. Este pulsador es una entrada directa al autómata, y mediante este accionamiento se procede a poner en marcha la periferia, requisito este necesario para poner en marcha el robot.



El accionamiento de este pulsador; y siempre y cuando no exista ninguna seguridad saltada, activa una marca de programa que está presente en cada uno de los movimiento de la instalación periferia + robot. Mediante la detección de flanco positivo de esta señal se pondrá en marcha la periferia, y una vez que soltemos este accionamiento, detectaremos el flanco de bajada de esta señal para poner en marcha el robot, si las demás señales de la Interface lo permiten.

%I1.1 -> Marcha -> Entrada directa del PLC.



- **Pulsador de Paro (10S1).** Mediante el Pulsador de paro, ejecutamos el paro total de la instalación, tanto periferia como el robot se detendrán de inmediato. Esta acción desactiva la marca de marcha general de programa, de manera que si la periferia no se encuentra en marcha, tampoco lo estará el robot. Bastará después volver a pulsar "**Marcha**" para que todo vuelva funcionar nuevamente ya que el paro de la instalación no implica ninguna seguridad activada.

%I1.0 -> Paro -> Entrada directa del PLC.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	



- **Conmutadores Permiso paletizar (10S13) y (10S14).**

Existen dos selectores en la instalación que determinan el permiso directo para paletizar tanto en la línea 1 como en la línea 2, estas serán condiciones directas que determinarán el permiso de descarga del robot. De esta manera si queremos que el robot no paletice en una de las líneas, activaremos este selector a la vez que se desactiva la barrera de seguridad correspondiente para permitirnos el paso a esa zona de trabajo si repercutir en la otra. Con el selector en la posición de prohibido paletizar, se deshabilita a través del PLC el permiso de descarga del robot en dicha línea, por lo que no entrará por programa en la zona de paletizado correspondiente.

Como se explicó en el capítulo de las seguridades, será necesario deshabilitar el permiso de paletizado en la línea cada vez que queramos entrar a por un palet, ya que de lo contrario saltarían las seguridades generales parando la otra línea que está en ejecución y tendríamos que realizar todo el proceso de rearme nuevamente.

Este accionamiento tiene por lo tanto dos propósitos:

1. Control sobre el permiso de paletizado en una de las líneas.
2. Deshabilitación de la barrera de seguridad de esa zona para recoger el palet.

%I1.13 -> Robot_No_Actúa_L1 -> Entrada directa del PLC.

%I1.14 -> Robot_No_Actúa_L2 -> Entrada directa del PLC.



- **Palet Nuevo de Línea (10S16) y (11S1).** Estos pulsadores serán necesarios accionarlos única y exclusivamente cada vez que se coloque un palet nuevo en la zona de paletizado, no actuando en ninguna otra secuencia de programa. Cada vez que el robot finaliza un palet, y el operario lo quita según corresponde, una vez situado el palet nuevo y vuelto el selector a su posición de paletizado, será necesario la confirmación mediante este pulsador para que el robot empiece

Nuevamente a paletizar sobre dicho palet. No obstante también será condición indispensable y necesaria que las fotocélulas de presencia **(2) y (3)** del sinóptico de la instalación, no detecten presencia de carga alguna.



En caso de existir presencia de carga en el palet nuevo, será necesario volver a girar el selector de línea, para entrar a quitar la carga, o para ajustar la fotocélula en caso de estar mal enfocada.

%I1.15 -> Palet_Nuevo_L1 -> Entrada directa del PLC.

%I3.0 -> Palet_Nuevo_L2 -> Entrada directa del PLC.



- **RESTOS** En este caso no se trata de una entrada directa al PLC, si no de una entrada directa al robot. Este elemento se accionará principalmente cada vez que se cambie de producto, de tipo de saco o de mosaico; es decir, cuando el operario no quiera finalizar el palet previamente configurado en cuanto a número de alturas y número de sacos. De esta manera,

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

accionado el botón de restos, el robot después de ejecutar cualquier movimiento de descarga que estuviera ejecutando, se detendría dando al PLC la señal de palet terminado como si de un palet completo se tratara.

En el **"SPS.sub"** o submit programa multitarea del robot (**Ver apartado 4.5.4.3 y ANEXO.A**), vemos estas líneas de programación:

```

;=====RESTOS=====
if $in[25]==true then
$out[25]=true
endif

```

De manera que tras la pulsación de este elemento, se activará una señal interna del robot **"\$OUT[25] = true"** que será leída posteriormente en el programa de paletizado, vej:

```

;=====RESTOS=====
continue
if $out[25]==true then
wait sec 1
pulse ($out[10],true,2)
pulse ($out[11],true,2)
saco_1=1
saco_2=1
base_data[3]=base_data[1]
base_data[4]=base_data[2]
wait sec 3
$out[25]=false
endif

;=====MANTENIMIENTO=====
continue
if (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false) and
(registro_1==2)) then
mantenimiento()
endif



;=====ZONA PALETIZADO 1=====
continue
if $out[25]==false then
continue
if (($in[11]==true) and ($in[13]==true) and ($out[30]==false)) then

```

**SECCION DEL PROGRAMA
MOSAICO_3 ()**

En esta parte del programa se ve que si la señal de restos esta activada, reseteamos las variables de saco, las bases, y el robot manda al PLC la señal de palet **finalizado en ambas líneas**. Seguidamente esperamos 3seg. Tiempo suficiente como para que el PLC se enteré de la finalización de la líneas, prohibiendo el permiso de descarga en ambas líneas hasta que no se sitúe un nuevo palet.

DI4-1 -> \$IN[25] -> Restos -> Entrada directa del robot.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

3.3.5 Terminal de diálogo XBT-N401.

Desde la pantalla visualizadora integrada, definiremos algunas opciones importantes de configuración y seleccionaremos el programa de paletizado que queremos ejecutar.

La pantalla que utilizaremos, es el terminal **XBT N401** de Telemecanique.





(Fig.3.3-18) Frontal y Trasero XBT-N401

Se trata de un terminal compatible con el protocolo **UNI-TELWAY**, de carácter alfanumérico retroiluminado con varios tonos diferentes (verde, rojo y Naranja) y pantalla de **4 líneas * 20 caracteres**. Alimentado a **24V**, este terminal tiene una capacidad interna de 512KB de memoria Flash, suficiente para cualquier aplicación sin necesidad de usar una memoria externa. La comunicación con el **TSXMicro** es a través de la Interface para autómatas de Schneider Electric: **Uni-Telway o Modbus**, siendo enlaces series punto a punto a través de un conector hembra tipo SUB-D de 25 contactos situado en la parte trasera.

Para navegar incorpora las teclas de servicio configurables para poder funcionar como teclas de control o enlace contextual.

◀, ESC, DEL, ▼, ▲, MOD, ENTER, ▶

- **ESC.** Anula una introducción, suspende o detiene una acción en curso, sube un nivel en un menú.
- **DEL.** Borra el carácter seleccionado en el modo introducción.
- **MOD.** Selecciona el campo variable que se va a introducir. Permite introducir datos en el campo siguiente, en cada pulsación, de izquierda a derecha y de arriba abajo.
- **ENTER.** Valida selecciones o datos introducidos y reconoce una alarma.
- ◀ ▶ Permite cambiar de página en un menú, visualizar las alarmas actuales, cambiar de dígito en un campo variable en el modo introducción o activar la función asociada a un enlace de función.
- ▼ ▲ Permiten subir y bajar en una página, seleccionar el valor de un dígito, seleccionar un valor en una lista de elección o aumentar y disminuir el

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

valor de un campo variable.

Junto con las teclas analizadas, también incorpora un piloto de alarmas, y dos flechas que se iluminan en el modo introducción de datos.



El software empleado para su configuración es el **XBT L100 o VIJEO DESIGNER LITE**, empleado para la creación de entornos de diálogo que controlarán el automatismo; y el cual, puede contener todo tipo de variables.



(Fig.3.3-19) Software Vijeo Designer Lite.

Las opciones configurables a las que tendremos acceso mediante dicha Interface serán las siguientes:

1. *Visualización y diagnóstico de los defectos principales relacionados con el funcionamiento y la marcha de la instalación.*
2. *Configuración de las líneas de paletizado.*
 - Selección del tipo de mosaico (1,2 ó 3).
 - Selección del número de alturas (mínimo 1, máximo 10).
 - Configuración del incremento de altura para la caída del saco.
 - Visualización del programa en ejecución.
3. *Visualización de los datos de producción de las dos líneas.*
 - Sacos totales.
 - Palet's terminados.
4. *Mandos directos sobre el robot.*
 - HOME.
 - MANTENIMIENTO.
 - LIBERACIÓN.

5. Ejecución de movimientos manuales.

- Movimiento manual de la cinta de saturación.
- Movimiento manual de los rodillos de recogida de saco.
- Movimiento manual del cilindro prensa.

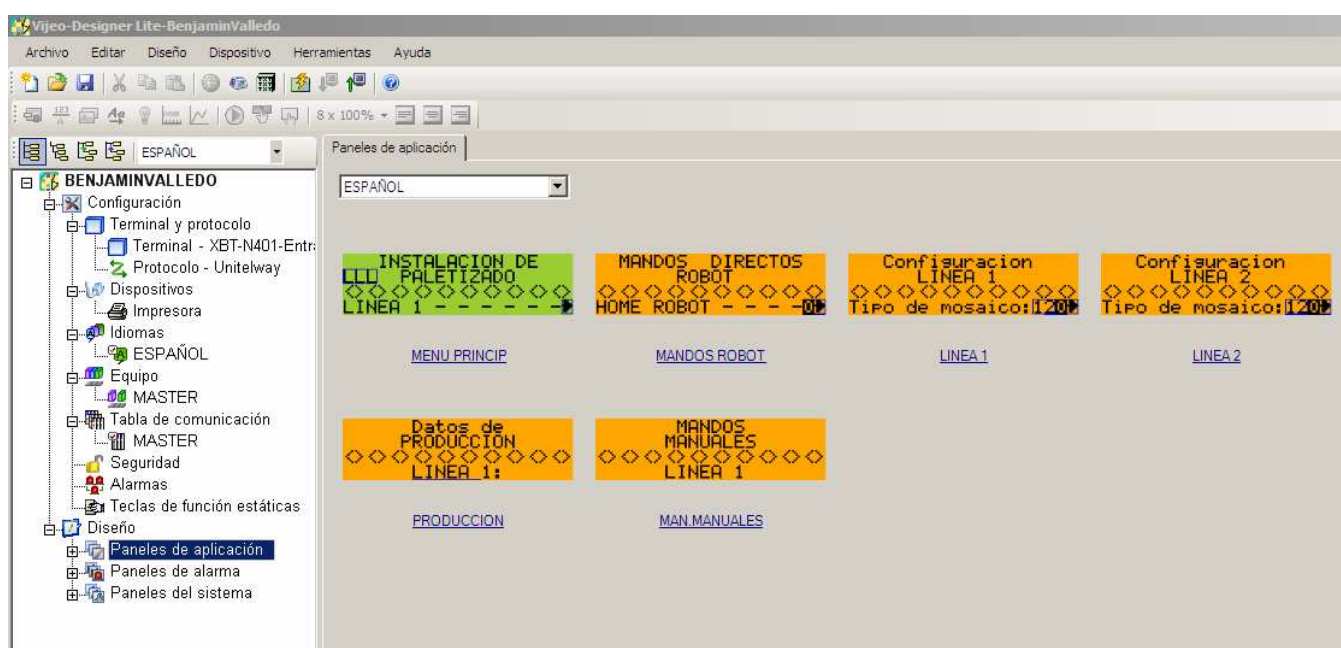
6. Acceso a páginas de sistema (NO UTILIZADO).

Cuando abrimos el programa desde el entorno de desarrollo **Vijeo-Designer-Lite**, tenemos acceso al menú principal de pantallas desde el que se nos mostrará todos los paneles a los que tendremos acceso una vez descargada la aplicación en el terminal (Fig.3.3-20).

El panel que se encuentra retroiluminado de color verde, será el menú principal desde el que accederemos a todas las pantallas de configuración. Las pantallas de configuración se encuentran de color naranja, y más adelante veremos como alarmas se presentan en color rojo.



NOTA: Para abordar el presente capítulo de forma satisfactoria sería recomendable la previa lectura sobre la configuración de Mosaicos en el capítulo Software de este proyecto.



(Fig.3.3-20) Entorno de desarrollo Vijeo Designer Lite.

Abordaremos a continuación las opciones destacadas.

3.3.5.1 Menú Principal.

Desde el menú principal se presenta el acceso al resto de paneles.



- **LINEA 1.** Visualizaremos y modificaremos las opciones de configuración de mosaico, altura e incremento de altura, lo que hagamos en LINEA 1 afectará a las dos líneas de paletizado, ya que en esta instalación las opciones de configuración serán las mismas para ambas líneas.

- **LINEA 2.** No se encuentra accesible, por la razón comentada anteriormente. Será un panel al cual no se tenga acceso. Sin embargo en

otras instalaciones la configuración de una LINEA de paletizado no tiene por que ser la misma, pero en nuestro caso solamente manejaremos las opciones de configuración de LINEA 1.

- **DATOS DE PRODUCCION.** Visualizaremos la cantidad de sacos puestos los palet's terminados.
- **ORDENES ROBOT.** Tendremos el control de ordenar al robot que ejecute ciertas maniobras mientras este se encuentra en automático. Veremos por ejemplo que para poder cambiar el programa de paletizado del robot, tendremos que mandarle primeramente a la posición HOME mediante esta opción.
- **MANDOS MANUALES.** Desde esta pantalla tendremos la posibilidad de realizar movimientos manuales en la cinta y rodillos de entrada, así como bajar el cilindro prensa.
- **ALARMAS.** Podremos visualizar si existe alguna alarma activada.

El resto de paneles pertenecen al propio sistema del XBT, y en principio no son de utilidad práctica para el manejo de la instalación.



Estando en el menú principal, navegaremos a través de las opciones comentadas mediante las flechas de navegación de desplazamiento arriba/abajo. Dos flechas de color verde nos indican con las teclas que podemos usar. De esta manera si vamos bajando irán apareciendo todos los paneles.

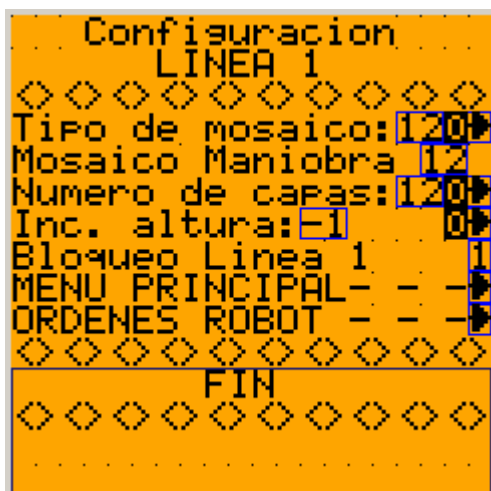


Cada vez que recorremos todos los paneles, una flecha intermitente nos dará la posibilidad de enlazar con la página que se encuentra seleccionada.



3.3.5.2 Página LINEA 1.



El menú correspondiente a la página seleccionada es el siguiente, y en donde realizaremos los siguientes ajustes:



- **Tipo de mosaico:** Dejaremos seleccionado el mosaico que queremos paletizar. Solamente podremos introducir en dicho campo los valores 1,2 ó 3. Más adelante explicaremos que condiciones son necesarias para que el robot coja el nuevo cambio de programa.
- **Mosaico Maniobra:** (Sólo lectura) Cuando el robot después de ir al HOME, adquiere realmente el programa a paletizar, este aparecerá en dicho campo. Muestra por lo tanto el programa de paletizado que se está

ejecutando realmente en el robot.

- **Número de Capas:** El número de capas se refiere realmente al número de alturas que se quiere paletizar en el mosaico seleccionado. El operario podrá seleccionar un mínimo de 1 altura, hasta un máximo de 10 alturas.
- **Inc.altura:** En este campo podremos introducir un incremento o decremento de la altura en cada capa. Este valor va desde +7 hasta -7 siendo cada unidad equivalente a 5mm. En los programas de paletizado tanto para los mosaicos de 3, 4 ó 5 sacos, se define un espesor del saco que será el incremento en el eje Z que el robot subirá por cada altura paletizada. Para el Mosaico de 5 tenemos por ejemplo lo siguiente:

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

DEF zona_1 ()

alt_incr_1=190

\$out[30]=true

coger_1 ()

INICIO DEL PROGRAMA

;;;saco 1 capa 1;;;;;;;;;;

if (saco_1==1) or (saco_1==11) or (saco_1==21) or (saco_1==31) or (saco_1==41) or (saco_1==51) or (saco_1==61) or (saco_1==71) then

PTP P1 CONT Vel= 100 % PDAT3 Tool[1] Base[3

ptp_rel {z -140} c_ptp

pulse (\$out[19],true,2)

wait sec 0.3

endif

;;;;;;;;saco 1 capa 2;;;;;;;;

.....

;;;;;;;;saco 5 capa 1;;;;;;;;

if (saco_1==5) or (saco_1==15) or (saco_1==25) or (saco_1==35) or (saco_1==45) or (saco_1==55) or (saco_1==65) or (saco_1==75) then

PTP P5 CONT Vel= 100 % PDAT10 Tool[1] Base[3

pulse (\$out[19],true,2)

wait sec 0.3

base_data[3].z=base_data[3].z+alt_incr_1+altura_1

endif

FINAL DEL PROGRAMA

;;;;;;;;saco 5 capa 2;;;;;;;;

if (saco_1==10) or (saco_1==20) or (saco_1==30) or (saco_1==40) or (saco_1==50) or (saco_1==60) or (saco_1==70) then

PTP P10 CONT Vel= 100 % PDAT11 Tool[1] Base[3

pulse (\$out[19],true,2)

wait sec 0.3


base_data[3].z=base_data[3].z+alt_incr_1+altura_1

endif

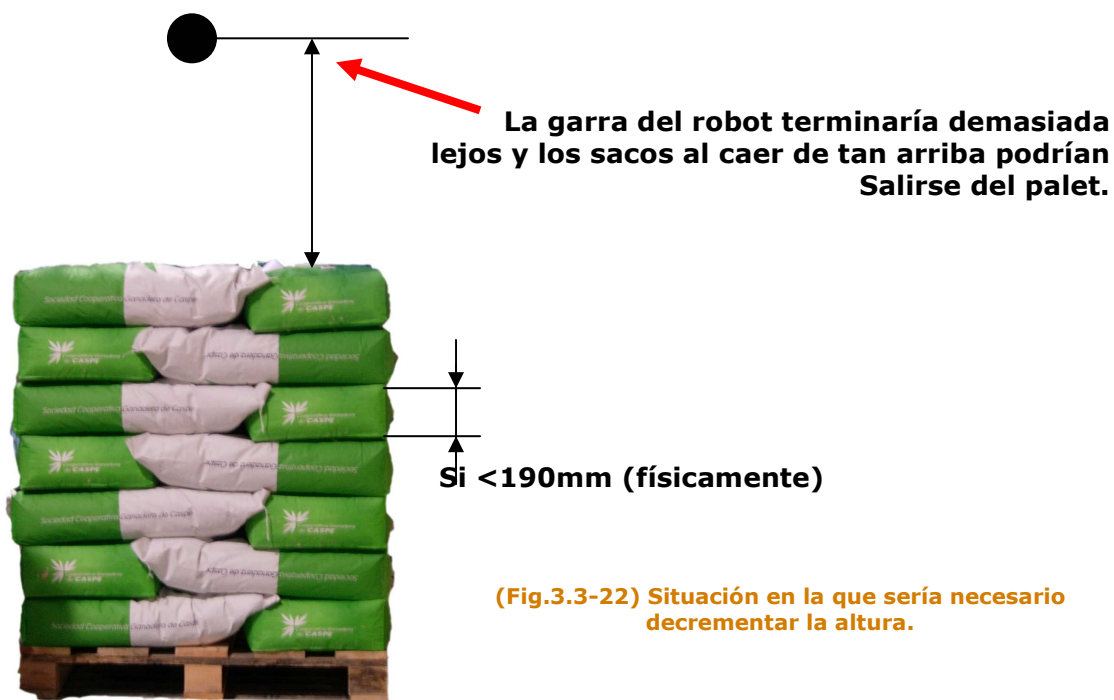
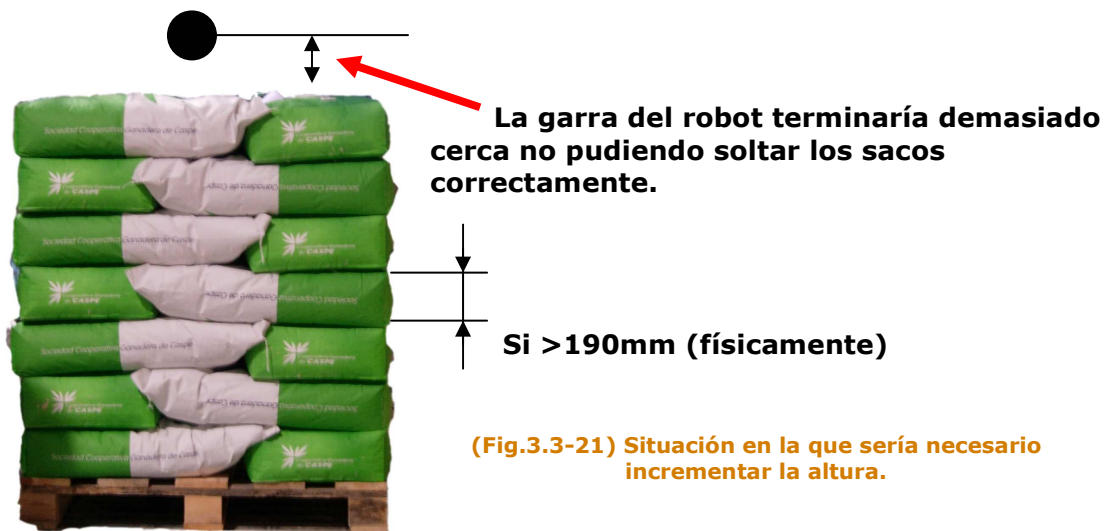
Cada vez que el programa entra en el subprogra de paletizado **Zona_1 ()**, hay una variable que es **alt_incr_1 = 190** donde se define el espesor del saco. De manera que cuando el robot ha depositado el saco numero 5 de la capa_1 o de la capa_2, la coordenada Z de la base a la que está referenciada dicho palet, se incrementa el valor **alt_incr_1 = 190**. De esta manera la próxima vez que el robot entre a depositar el primer saco de la siguiente altura, lo hará 190mm más arriba.

base_data[3].z=base_data[3].z+alt_incr_1+altura_1

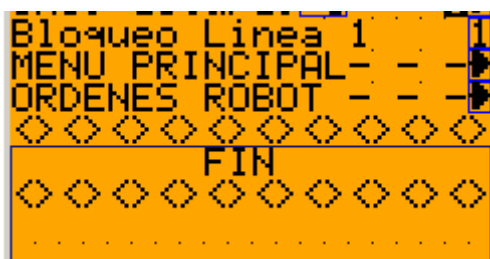
Existe otra variable en la línea de programa denominada **altura_1** esta será la variable que tome el valor configurado en pantalla, de manera que podemos hacer que el robot incremente o decremente su altura $\pm 7\text{mm}$. Este variable contribuye ajustando la altura del robot a la que este deja caer el saco al palet, ya que aunque la medida establecida para dicho saco sea 190mm, puede ser que los sacos no venga tan llenos y tengamos que decrementar un poco la altura, ya que si no, en la última altura de paletizado el robot estaría soltando el saco desde muy arriba, con lo que la caída del saco podría desplazarse al golpear con la capa de abajo, o puede ser que este demasiado lleno y tengamos que incrementarla ya que

ROBITEK.System Dpto.Robótica	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

si no en la última altura, el robot resultaría estar demasiado cerca para dejar caer el siguiente saco Figuras (Fig.3.3-21 y 3.3-22).



Incrementando el valor entre $\pm 35\text{mm}$, en cada altura, podemos variar la posición de la garra del robot hasta $\pm 350\text{mm}$, que serían las 10 alturas como máximo multiplicado por siete y por cinco.



Las siguiente Opción **Bloqueo Línea1** carece de funcionalidad puesto que el bloqueo de las zonas de paletizado lo haremos mediante los selectores del armario. Mediante **MENU PRINCIPAL** y **ORDENES ROBOT** tendremos acceso a dichas páginas.

Para poder modificar el mosaico a paletizar o el número de capas, tendremos que situarnos con las flechas de navegación, hasta visualizar la línea **Tipo de Mosaico** o **Número de capas** como muestra la figura.

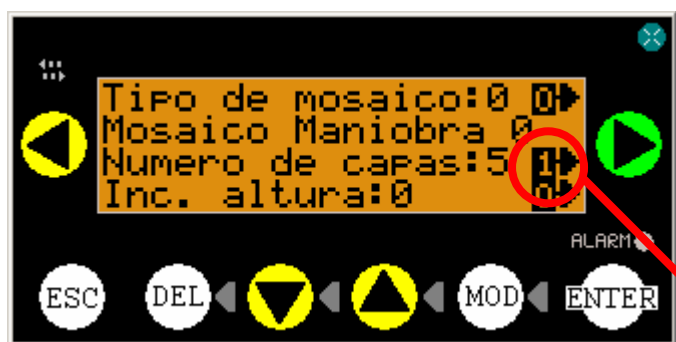


Mediante la tecla **MOD** seleccionaremos la línea que queremos modificar, de manera que un cursor aparezca intermitente. En este caso se encuentra en **Número de capas** por lo que el valor que modificaremos serán las alturas. Para incrementar o decrementar el campo seleccionado, usaremos flecha

arriba o flecha abajo. En el caso de las alturas el valor máximo que podremos poner será de 10, por lo que si tratamos de poner un valor mayor, nos aparecerá un mensaje de rebasamiento del valor.





Mediante **ESC**. Saldremos de la pantalla y volveremos para introducir un valor coherente, por ejemplo 5.



Una vez tengamos marcado el valor, lo validaremos mediante **ENTER**. Después nuevamente con flecha arriba y flecha abajo nos situaremos en la línea de **Numero de capas** hasta que el valor marcado como cero se ponga a parpadear.



Para enviar este nuevo valor al PLC que gestiona la comunicación con el Robot, tendremos que marcar en flecha derecha para enviar el valor al autómat, esto se representa mediante un **"1"** en el campo que antes ponía un cero, haciendo entender al operario que el valor se ha registrado correctamente y ha sido enviado al PLC. El **"1"** se mantendrá mientras mantengamos pulsada la tecla derecha.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

De la misma manera procederíamos en caso de querer cambiar tanto el número de mosaico como el valor de incremento de altura.

La comunicación en esta pantalla con el PLC se hace a través de variables directas que podemos consultar en los ANEXOS de programa; así por ejemplo:

%MW1 -> Tipo de mosaico.
%MW2 -> Numero de capas.
%MW3 -> Incremento de altura.
%MW18 -> Mosaico Maniobra o seleccionado.

1->  **%MW20:X0 -> Confirmación para cambiar tipo de Mosaico**

1->  **%MW20:X1 -> Confirmación para cambiar Capas.**

1->  **%MW20:X2 -> Confirmación para cambiar incrementos.**

3.3.5.3 Página Datos de producción.



Este será un Menú de solo lectura, puesto que en él simplemente veremos datos de producción sin la posibilidad de ejercer ningún control, salvo la puesta a cero de los contadores

Los datos productivos serán los mostrados en la información de LINEA 1 para las dos zonas de paletizado.


- **Sacos:** Contador que muestra el número total de sacos que deja el robot sobre el palet de la línea_1 + línea_2.
- **Palets:** Contador que muestra el número de palet's finalizados por el robot. Se considerará palet finalizado, cualquier tipo de palet que haya finalizado por configuración de alturas, o mediante el botón de **RESTOS**.
- **Puesta a cero:** Mediante el método mostrado anteriormente pondremos a cero los contadores anteriores.

Cuando estemos dentro de esta pantalla se nos mostrará la pantalla retroiluminada de color Naranja. La forma en que se ponen a cero los contadores, será posicionándonos sobre la línea Puesta a cero hasta que veamos la intermitencia en el simbólico cero. Simplemente tendremos que pulsar sobre la tecla de función para transmitir el bit al autómatas.



En este caso la comunicación con el PLC, se realizará por medio de las siguientes variables:

%MW8 -> Sacos Totales.
%MW9 -> Palet's Totales.

1->  **%MW20:X6 -> Reset de contadores.**

3.3.5.4 Página Ordenes Robot.



La Pantallas mandos del Robot, nos permiten ejercer una acción directa sobre el control del robot. Siempre a través del autómata y si las condiciones de programa son las correctas, podremos observar sus efectos de manera inmediata.



- **HOME ROBOT.** Esta opción llevará al robot a su punto HOME o de inicio. También será la posición a la que irá el robot cuando seleccionemos el programa de ejecución **CELL.src** para el manejo mediante automático externo. La explicación a esto es que para modificar el programa de paletizado o seleccionar un programa de paletizado la primera vez que ejecutamos cell.src el robot ha de encontrarse en esta posición de seguridad. Dicha posición se alcanza mediante el subprograma **Home ()**, y dentro se activa la señal que necesita el PLC para mandar los datos de paletizado al robot (Número de mosaico, y número de capas).

Para los datos sobre el incremento de altura no es necesario que el robot se encuentre en el HOME, pudiéndose cambiar en cualquier momento del programa de paletizado.

Cuando se inicia **CELL.src** por primera vez, el robot irá a HOME sin necesidad de que lo seleccionemos desde la pantalla, ya que desde cell se entrará al programa de paletizado que este configurado en pantalla, pero para ello antes ha sido preciso cargar los datos. De hay que el primer movimiento que hará el robot al reiniciar el sistema en **Arranque en Frio**, sea el de HOME.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

Si queremos cambiar el programa de paletizado una vez que el sistema fue iniciado (normalmente es lo que ocurrirá siempre), la entrada a HOME estará condicionada a que los palet's de las líneas estén vacíos; es decir, que no haya empezado a paletizar con el mosaico anterior. Esto se consigue no permitiendo la entrada de sacos, de manera que el robot termine con los sacos de entrada para después mediante el pulsador de restos, finalizar el palet, o incluso bloqueando una línea para que termine el paletizado por la otra, etc. Al no haber sacos, el robot quedará parado, y en este momento mediante la pantalla llevaremos el robot al pto. HOME. Generalmente el robot podrá ir a HOME, si ha finalizado los palet's de ambas líneas y no ha iniciado uno nuevo, o cada vez que hagamos un restos a las dos líneas, posibilitando la entrada a HOME desde esta pantalla.

Dentro de HOME, el robot ejecutará una coincidencia de trayectoria y realizará dos movimientos arriba y abajo, para indicar al operario que está en HOME, seguidamente el robot se quedará parado a la espera de recibir los datos que configuremos desde la pantalla.

15.1 "CELL.SRC"

```

DEF CELL ( )
EXT mosaico_5 ( )
EXT mosaico_4 ( )
EXT mosaico_3 ( )
EXT home ( )

:FOLD INIT
DECL CHAR DMY[3]
DMY[]=""
:ENFOLD (INIT)
:FOLD BASISTECH INI
GLOBAL INTERRUPT DECL 3 WHEN $ST
INTERRUPT ON 3
BAS (#INITMOV,0 )
:ENFOLD (BASISTECH INI)
home ( )
:FOLD CHECK HOME
$H_POS=XHOME
IF CHECK_HOME==TRUE THEN
  P00 (#CHK_HOME,#PGNO_GET,DMY[
ENDIF
:ENFOLD (CHECK HOME)
:FOLD PTP HOME Vel= 100 % DEFAULT
1:PTP, 2:HOME, 3:, 5:100, 7:DEFAULT
$H_POS=XHOME

```

(Fig.3.3-23) Entrada a HOME en CELL.src

19.4 HOME.src

```

DEF home( )
coi_sgrd ( )

:marca para saber que estamos dentro del Home y cambiar de programa
$out[21]=true

wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false)
and (registro_1==3))

:=====CAMBIO DE MOSAICOS=====

:Mosaico seleccionado
$out[21]=false

:=====CAMBIO DE CAPAS=====


capas_1=reg_capas_1
if program==1 then
  sacos_1=capas_1*5
  sacos_2=capas_1*5
endif
if program==2 then
  sacos_1=capas_1*4
  sacos_2=capas_1*4
endif
if program==3 then
  sacos_1=capas_1*3
  sacos_2=capas_1*3

```

Una vez que estemos dentro de HOME. Lo primero que se ejecuta es el subprograma **coi_sgrd ()** donde en función del eje principal **A1** el robot realiza una coincidencia de trayectoria segura. Seguidamente pondrá a **true** la salida **\$OUT[21]** para que el PLC transmita el nuevo valor de la variable PROGRAM definida en el **\$CONFIG.dat (Ver Variables del robot)**, después el robot quedará parado esperando la condición de salida, que explicaremos más detalladamente en la sección del software.

wait for ((\$in[17]==false) and (\$in[18]==false) and (\$in[19]==true) and (\$in[20]==false) and (registro_1==3))

En esta línea de programa se espera a que haya una transferencia de datos mediante las señales **(((\$in[17]==false) and (\$in[18]==false) and (\$in[19]==true)**

	Proyecto	PROYECTO FIN DE CARRERA
	Documento	Instalación de Robot Paletizado.



and (\$in[20]==false). De esta manera en el programa multitarea **Submit.sps** se realiza la captura del valor de capas y se guarda en una variable definida en el **\$Config.dat** denominada **reg_capas_1**. Mientras que en la Variable **SIGNAL PROGRAM** también definida en **\$Config.dat** se guarda el número de mosaico seleccionado enviado al PLC a través de la pantalla.

```

;=====CAPTURA DE DATOS BINARIOS=====

//////////Capas
if (($in[15]==true) and ($in[17]==false) and ($in[18]==true) and ($in[19]==false) and
($in[20]==false)) then
reg_capas_1=registro_1
endif

```

Para sacar al robot del modo de espera ha de cumplirse la condición de la función **"Wait for"**, esto se consigue mediante la orden del robot **LIBERACION** también desde la pantalla **XBT**. De esta manera antes de liberar al robot, hemos tenido que enviarle al HOME, transferir desde el XBT N401 el número de mosaico y el número de capas y posteriormente liberar.

En el momento que se cumpla la condición **Wait For**, el robot en función del programa seleccionado y el número de capas, establecerá el número de sacos que ha de paletizar por palet.

```

capas_1=reg_capas_1
if program==1 then
sacos_1=capas_1*5
sacos_2=capas_1*5
endif
if program==2 then
sacos_1=capas_1*4
sacos_2=capas_1*4
endif
if program==3 then
sacos_1=capas_1*3
sacos_2=capas_1*3

```

La variable **sacos_1** y **sacos_2** será la variable que determinará cuando se ha llegado al final del palet, mientras que otra variable **saco_1** y **saco_2** se va incrementando en cada zona de paletizado por cada saco depositado.

Ejecutada esta parte de código y con la variable **Numero de PROGRAM** cargado con el programa actual, el robot entrará en el programa de paletizado correspondiente dentro del cell, o saldrá del programa que estaba ejecutando para entrar en cell y cambiarlo.



```

;ENDFOLD (AUTOEXT INI)
LOOP
P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
SWITCH PGNO ; Select with Programnumber
CASE 1
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset
Progr.No.-Request
mosaico_5 ( ) ; Call User-Program
CASE 2
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset
Progr.No.-Request
mosaico_4 ( ) ; Call User-Program
CASE 3
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset
Progr.No.-Request
mosaico_3 ( ) ; Call User-Program
;DEFAULT
;P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
ENDSWITCH
ENDLOOP

```

Dentro de CELL se entra en el programa de paletizado correspondiente. **Mosaico_5()**, **mosaico_4()** o **mosaico_3 ()**. Según el valor de los bit's **PGNO_Bit_1**, **PGNO_Bit_2** y **PGNO_Bit_3** (Ver Sección Automático externo).

Veamos a continuación

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

las últimas líneas de programa **MOSAICO_3 ()** para ver que sería necesario que se cumpliera en caso de que queramos cambiar de programa.

```

;=====SALIDA CONDICIONAL=====
continue
if (((program==1) or (program==2)) and ($in[11]==false) and (saco_1<2) and (saco_2<2))
then
wait sec 1
exit
endif

endloop
wait sec 0
coi_sgrd ()

END

```

**FINAL DE PROGRAMA
MOSAICO_3 ()**

Decir primero que:

PROGRAM = 1 -> Se corresponde con MOSAICO_5 ()
PROGRAM = 2 -> Se corresponde con MOSAICO_4 ()
PROGRAM = 3 -> Se corresponde con MOSAICO_3 ()

De esta manera si estamos dentro de MOSAICO_3 () ; es decir, PROGRAM = 3, y hemos provocado la entrada en **Home ()** dentro de MOSAICO_3 (), si hemos seleccionado un programa distinto **program == 1 o program == 2** y los palet's están vacíos **saco_1<2 y saco_2<2** y no tenemos permiso de carga; es decir, no están entrando sacos al alimentador de rodillos **\$in[11] = false**, entonces se ejecutará la salida del bucle infinito mediante la llamada **exit**, se finalizará dicho bucle **endloop**, se ejecutará una coincidencia de trayectorias nuevamente por seguridad, de manera que el robot se mueva arriba y abajo indicando el estado, y al finalizar MOSAICO_3 (), el programa entrará en el bucle de **CELL.src** para entrar en el nuevo programa de paletizado.

```

;ENDFOLD (AUTOEXT INI)
LOOP
P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
SWITCH PGNO ; Select with Programnumber

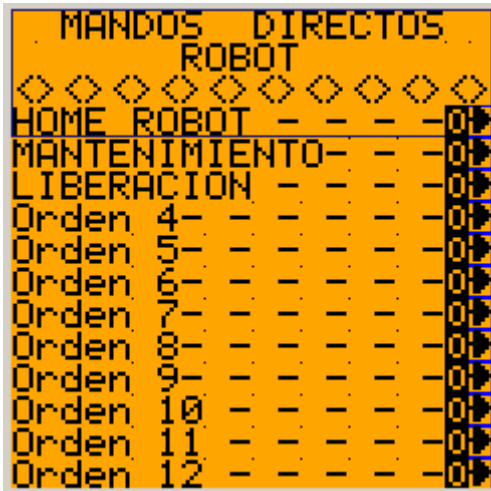
CASE 1
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_5 ( ) ; Call User-Program

CASE 2
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_4 ( ) ; Call User-Program

CASE 3
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_3 ( ) ; Call User-Program

;DEFAULT
;P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
ENDSWITCH
ENDLOOP

```



- **MANTENIMIENTO.** Mediante esta opción, que podremos ejecutar en cualquier momento estando dentro de cualquier programa de paletizado Mosaico_3 (), mosaico_4 () o Mosaico_5 (), el robot ejecutará una coincidencia de trayectorias por seguridad y se desplazará hasta un punto de acceso seguro donde se detendrá y esperará a ser liberado. Este punto de acceso seguro se empleará principalmente para el mantenimiento o reparaciones de la garra del robot por lo tanto la garra del robot se encontrará en una

posición libre de obstáculos a la vez que accesible.

```
DEF mantenimiento( )
INI
;Make your modifications here

coi_sgrd ()

PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P2 CONT Vel= 30 % PDAT2 Tool[1] Base[0]
PTP P3 CONT Vel= 30 % PDAT3 Tool[1] Base[0]
;punto de mantenimiento
PTP P4 CONT Vel= 30 % PDAT4 Tool[1] Base[0]

;=====salida de mantenimiento=====

wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false)
and (registro_1==3))



PTP P5 CONT Vel= 30 % PDAT5 Tool[1] Base[0]

coi_sgrd ()

END
```

La entrada a MANTENIMIENTO no está condicionada a que los palet's estén vacíos y en general la instalación; será una orden que podemos ejercer en cualquier momento, a lo que hay que entender que si el robot está cogiendo un saco o soltando un saco en el palet, en el momento de mandar la orden, el robot siempre finalizará primero la acción de carga y descarga del saco antes de irse a la posición de mantenimiento. Vemos en el programa que después de ejecutar el subprograma de coincidencia de trayectorias, el robot se desplaza a la velocidad reducida del **30%** hasta el punto **PTP4**, punto donde el robot se detendrá hasta nueva orden **wait for ((\$in[17]==false) and (\$in[18]==false) and (\$in[19]==true) and (\$in[20]==false) and (registro_1==3))**.

Una vez situado en la zona de mantenimiento, el operario de podrá parar la instalación, hacer saltar las seguridades, apagar los armarios eléctricos, para así asegurar de que el robot en todo momento no ejercerá ningún tipo de movimiento

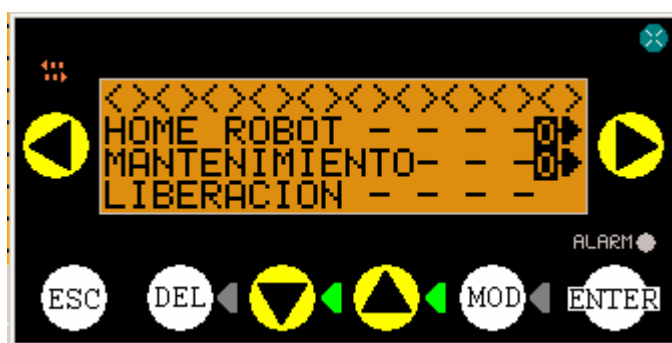
	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

durante las operaciones de mantenimiento. Se recomienda para esto pulsar la seta de emergencia dispuesta en la KCP para tales circunstancias.

Después de realizar las operaciones de mantenimiento oportunas, la instalación tendrá que rearmarse en el caso de que las seguridades no estuvieran rearmadas, seguidamente y al igual que ocurría con la posición **HOME** del robot, tendremos que liberar de la posición de mantenimiento mediante la pantalla XBT. La liberación desde la pantalla, provoca que la variable **registro** adquiera el valor de '3'; es decir, **registro_1 = 3**, de esta manera se producirá la liberación del robot ejecutando nuevamente las coincidencias de trayectoria, y prosiguiendo posteriormente con la ejecución del programa de paletizado.




- **LIBERACIÓN.** Accionamiento al cual hemos hecho referencia en esta sección, se utilizará en los siguientes casos:
 1. Para liberar al robot de la posición HOME haya cambio o no de programa.
 2. Para liberar al robot de la posición MANTENIMIENTO.

La pantalla en esta página será la ilustrada a continuación, en la que simplemente nos dedicaremos a enviar al PLC las órdenes mediante la activación de un bit, en función de si la selección ha sido HOME, MANTENIMIENTO o LIBERACIÓN.





Nos situaremos en la línea de pantalla correspondiente mediante las flechas de navegación, hasta que el campo "0" deseado empiece a parpadear y enviaremos la orden mediante flecha derecha, mientras

observamos como el campo "0" pasa a ser "1" indicándonos el envío de dicha orden. Las señales de comunicación con el autómatas serán las siguientes:

- 1->  **%MW21:X0 -> Orden ir a HOME.**
Registro_1 = 1;
- 1->  **%MW21:X1 -> Orden ir a MANTENIMIENTO.**
Registro_1 = 2;
- 1->  **%MW21:X3 -> Orden LIBERACION.**
Registro_1 = 3;

El resto de órdenes que podríamos encontrar en la pantalla son de reserva, por lo tanto no se ejercerá ninguna instrucción ni mando posible sobre la instalación. Así mismo, cada vez que ejecutamos una orden el PLC modificará la

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

variable de programa del robot "**Registro_1**" tal cual se muestra, de manera que se cumplan o no las condiciones necesarias en el programa.

3.3.5.5 Página Mandos Manuales.



Cada vez que accedamos a los movimientos manuales, se nos mostrará una pantalla como la de la figura. Desde aquí podremos realizar un mando manual sobre los movimientos tanto de la cinta como de los rodillos de entrada de saco de la instalación, como de la subida y bajada del rodillo prensa. La activación de movimientos manuales, nos ayudan a ejecutar maniobras y comprobar el buen funcionamiento de los motores de la


instalación.

- **CINTA DE ENTRADA.** Activaremos el movimiento manual de la cinta de saturación. Dicho movimiento se puede hacer con la instalación en marcha sin ser necesario ningún paro.
- **MESA DE RECOGIDA.** Activaremos el movimiento manual de los rodillos de alimentación. Dicho movimiento se puede hacer con la instalación en marcha sin ser necesario ningún paro.
- **BAJAR RODILLO.** Tendremos posibilidad de bajar el rodillo prensa, para realizar los ajustes que sean necesarios.



1->  **%MW22:X0 -> Activar movimiento cinta de entrada.**

1->  **%MW22:X2 -> Activar movimiento mesa de recogida.**

1->  **%MW22:X3 -> Activar movimiento cilindro.**

3.3.5.6 Página de Alarmas.

Las páginas de alarma retroiluminada en color rojo, indican cualquier fallo en el proceso y su ventaja reside en su visualización activada por eventos. Estos eventos aparecen de manera automática de producirse alguno de ello y siempre podremos cancelar la página de alarmas mediante la tecla de función “ESC”.

1. Durante el funcionamiento

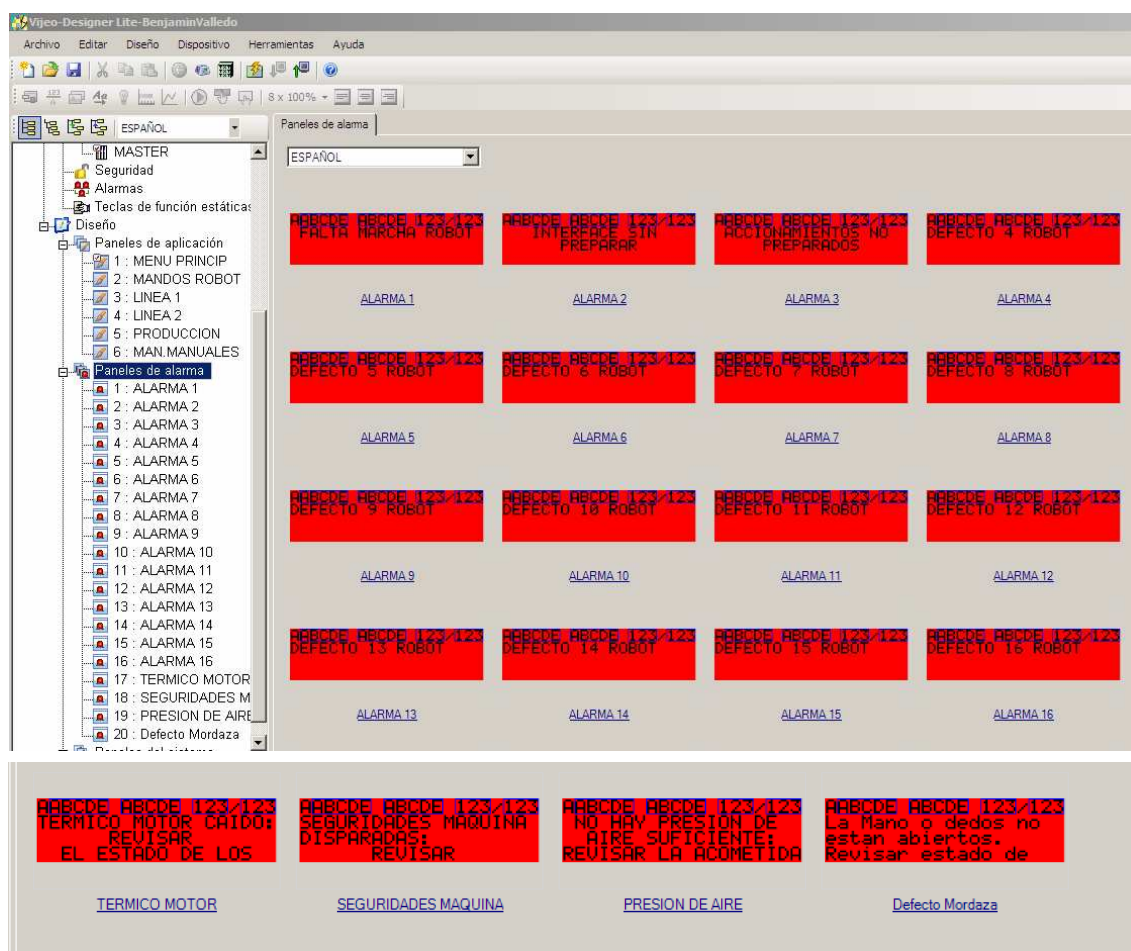
Cuando se produce un fallo, a menudo se debe a otros fallos. Los niveles de prioridad permiten visualizar en el terminal el fallo más importante, el fallo que presenta el máximo riesgo para el proceso. La aparición de cualquier fallo se marca con la fecha y la hora.



2. Durante el mantenimiento

El terminal memoriza los fallos en una secuencia (registro) facilitando así la búsqueda del motivo del fallo.

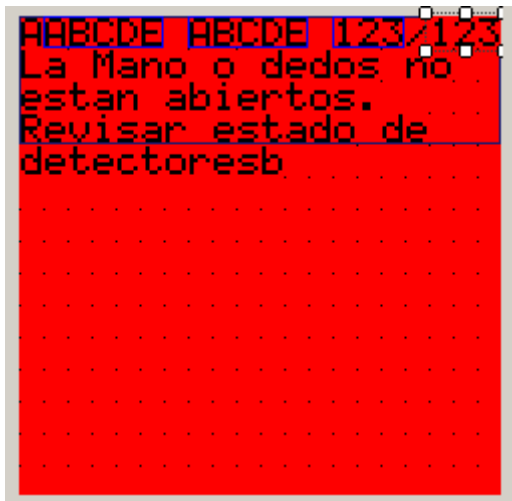
En el menú de alarmas desde el “Vijeo Designer Lite” presenta la distribución de las alarmas y eventos programados en la instalación. Se presentarán a continuación aquellas que tienen un impacto directo sobre el sistema de robot paletizado.

(Fig.3.3-24) Alarmas del Terminal XBT.



	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

- **Defecto Mordaza.**



El defecto de mordaza tal como hemos descrito a lo largo de la presentación y sinóptico de la instalación, se produce cuando el robot, o los sensores de la mordaza del robot, no detectan la posición de reposo una vez depositado el saco en el palet. Recordemos que la posición de reposo es "Dedos abiertos" y "Manos abiertas". De esta manera, si al cabo de un tiempo estimado de 4 seg. Desde que el robot soltó el saco y se posicionó en el punto de espera, los detectores de mordaza no detectan la posición abierta, el robot activará su salida de defecto y se lo transmitirá al PLC que hará saltar la alarma en la pantalla. Veamos pues cual sería la

Interface de esta comunicación:

1. **\$IN[27]** -> Entrada del robot a través del sensor magnético de la mordaza que indica **"Dedos abiertos"**.
2. **\$IN[28]** -> Entrada del robot a través del sensor magnético de la mordaza que indica **"Manos abiertas"**.
3. **\$OUT[12]** -> Salida del robot cuyo estado es **"true"** cuando han transcurridos los 4seg. Sin detectar la posición de reposo.
4. **%I5.11** -> Entrada al PLC que llega mediante la señal del robot **\$OUT[12]**.
5. **%MW119:X3** -> Bit de alarma que activará el PLC para su aparición en la pantalla.

El defecto de mordaza se evalúa en los subprogramas de paletizado **"Zona_1 () y Zona_2"** para el mosaico de 5, **"Zona_11 () y Zona_22 ()"** para el **mosaico** de 4 y **"Zona_111 () y Zona_222 ()"** para el mosaico de 3. Su sintaxis sería la siguiente:

```
;;;;;;;;;;;;punto elevado encima de zona 2
```

```
SYN PULSE 17 'Abrir_Manos' State= TRUE Time= 0.5 sec at START Delay= 0 ms
```

```
PTP P16 CONT Vel= 100 % PDAT16 Tool[1] Base[0
```

```
PTP P11 CONT Vel= 100 % PDAT1 Tool[1] Base[0
```

```
PTP P24 CONT Vel= 100 % PDAT24 Tool[1] Base[0
```

```
PTP P25 CONT Vel= 100 % PDAT25 Tool[1] Base[0
```



```
;verificacion manos y dedos abiertos
```

```
continue
```

```
loop
```

```
continue
```

```
if (($in[27]==false) or ($in[28]==false)) then
```

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

```

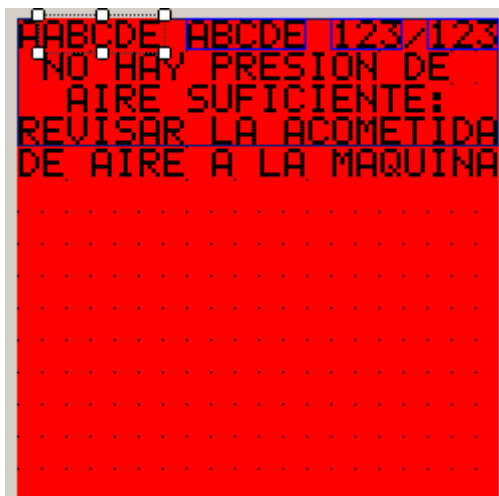
$timer_stop[1]=false
endif
continue
if $timer[1]>=4000 then
wait sec 0
$timer_stop[1]=true
$timer[1]=0
pulse ($out[12],true,2)
endif
continue
if (($in[27]==true) and ($in[28]==true)) then
$timer_stop[1]=true
$timer[1]=0
exit
endif
endloop
END

```

El robot antes de dejar el saco realiza la apertura de dedos y después de dejar el saco, realiza la apertura de las manos mediante la función **Synpulse**. Seguidamente ejecuta unos movimientos para situarse en su posición de espera de entrada de sacos. Una vez posicionado se evalúa la posición de reposo de la mordaza, para ello se genera un bucle "**Loop**" en el que el robot no saldrá nunca mientras la mordaza no se encuentre en reposo. Dentro de dicho bucle se evalúan las señales de los magnéticos y se va incrementando un tiempo en caso de que alguna de ellas no detecte su posición de reposo. Mediante "**\$timer_stop[1] = false**" se activa el temporizador para que inicie su cuenta, si su lectura es mayor de 4000mseg. Se activará la salida de defecto.



En caso de que la posición de reposo haya sido adquirida, el temporizador es detenido "**\$timer_stop[1] = true**" y es puesto a cero por si acaso **\$timer[1]=0**, finalizándose el bucle y el subprograma de zona, por lo tanto volvemos al programa de paletizado en ejecución esperando la señal de permiso para cargar nuevo saco.

- **Defecto Presión de aire**



El defecto de presión de aire es activado por la señal de presostato incorporado en la unidad de mantenimiento que hay en la entrada del robot, tal como vimos anteriormente. Es importante evaluar dicho defecto y parar la instalación. Ya que en caso de falta de aire, los elementos móviles de la mordaza no funcionarían correctamente, por lo que podría caerse un saco al no existir presión de sujeción suficiente.

La Interface de señales que se manejan en este caso es la siguiente.

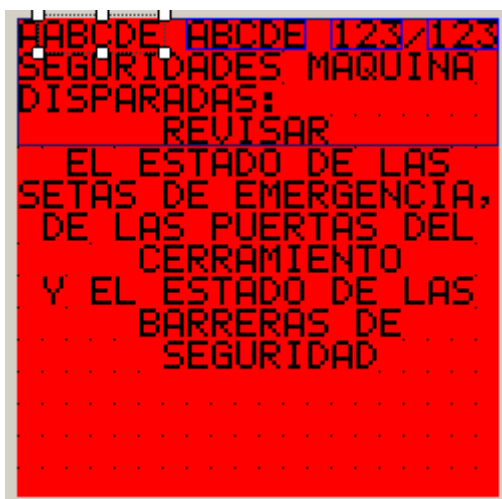
	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

1. **%I1.5** -> Entrada al PLC que llega mediante la señal del presostato.

2. **%MW119:X2** -> Bit de alarma que activará el PLC para su aparición en la pantalla.

La señal del presostato se evalúa con lógica negativa; es decir, si esto correcto es que esté marcando señal mediante "true" de manera que si esta señal cae a "**false**" se activará una marca de programa del PLC, ejecutando el paro de la instalación y provocando el paro del robot. En este caso será necesario esperar para que la presión sea correcta y poder poner en marcha la instalación mediante el accionamiento adecuado del armario eléctrico.

- **Seguridad Maquina**



Las seguridades de máquina provocan el paro inmediato de la instalación. Esta alarma está provocada por la desactivación del módulo de seguridad; provocado por lo tanto, por la activación de alguna de las setas de emergencia o por saltarnos la barrera de seguridad mediante una maniobra incorrecta.

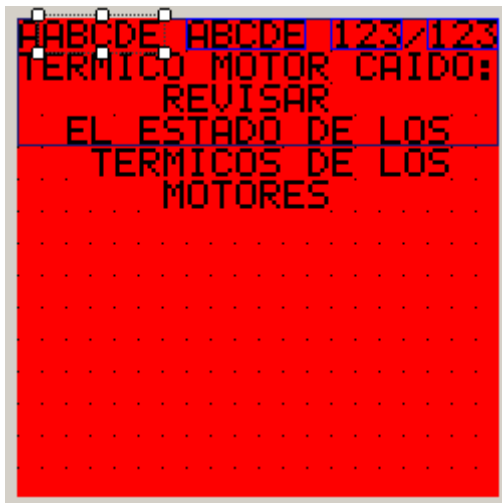
Dicha alarma provoca la caída de potencia de la instalación, y la marcha de la misma, imposibilitando el rearme del robot hasta que esta no sea solventada.

1. **%I1.4** -> Entrada al PLC que llega mediante el contacto abierto del contactor **7KM2** de seguridad.

2. **%MW119:X1** -> Bit de alarma que activará el PLC para su aparición en la pantalla.

La única manera de rearmar este defecto, es mediante la activación el módulo de seguridades tal como se vio en el capítulo correspondiente, para después proseguir con el rearme del robot y la marcha de la instalación.

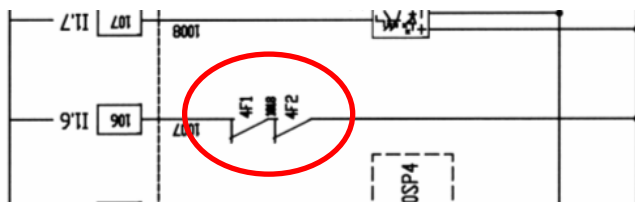
- **Térmico motores**



El defecto térmico motores, revisa el estado de las seguridades térmicas de los dos motores de la instalación; siendo estos, el motor de la cinta de saturación y el motor de los rodillos de alimentación.

Ambos térmicos poseen un **contacto abierto** que permanece cerrado si el térmico está OK (Fig.3.4-25). De esta manera los dos contactos de los térmicos se encuentran seriados a una entrada del autómata de manera que cualquiera de ellos que salte activará la alarma.

La lógica de funcionamiento es la misma que para El caso del defecto de presión, pues en caso de producirse se para la instalación general, y si esta no se encuentra en marcha se detiene el paro del robot, siendo la única manera de volverla a poner en marcha el cierre de los contactos.



CELULA PRESENCIA



SERIE TERMICOS

(Fig.3.3-25) Seguridades Térmicas.

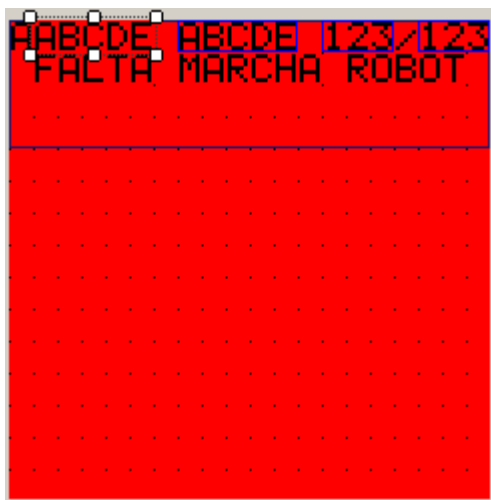


1. **%I1.6** -> Entrada al PLC que llega mediante la serie de contactos cerrados de los dos térmicos.
2. **%MW119:X0** -> Bit de alarma que activará el PLC para su aparición en la pantalla.

La asignación de las siguientes alarmas que van desde **Alarma_1** hasta **Alarma_3** son eventos informativos sobre el estado de la marcha del robot, que no implican la activación de ninguna seguridad importante de la instalación.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

- **Alarma 1 (Falta Marcha del Robot).**

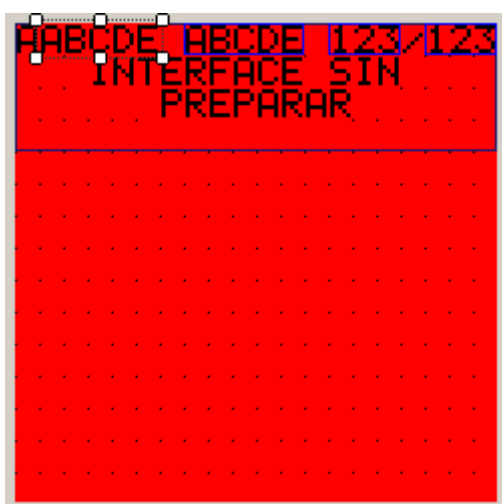


Dicho evento estará presente mientras el robot no se encuentre en marcha. Se considerará la marcha del robot en este caso, si tiene habilitado el movimiento; es decir, si la instalación del robot está en marcha, y además está ejecutando un programa. De esta manera si toda la periferia se encuentra en marcha a falta del robot, se notificará en pantalla mediante dicho mensaje. Esto puede ser si el robot no tiene ningún programa de paletizado seleccionado; es decir, que no se va a mover ni va hacer nada.

Las señales implicadas son las siguientes:



1. **%I5.5 -> Di_APPL_RUN** Entrada al PLC que llega desde el robot indicando que hay algún programa ejecutándose.
2. **%Q6.6 -> Do_Move_Enable** Salida del PLC habilitando el movimiento del robot
3. **%MW118:X0 -> Bit de alarma** que activará el PLC para su aparición en la pantalla.

- **Alarma 2 (Interface sin preparar).**



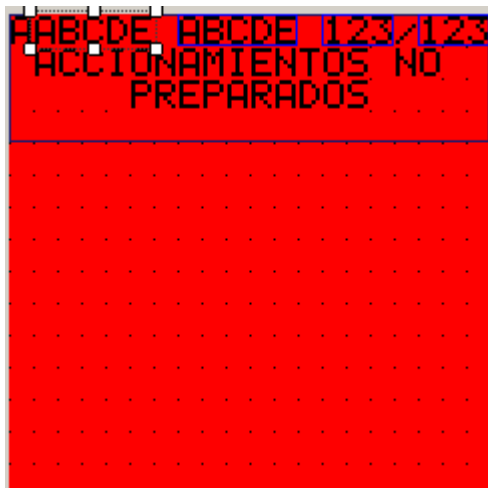
El siguiente evento, sólo se notificará en caso de que la Interface de comunicaciones del PLC con el Robot, no se encuentra establecida o no puede ser establecida. Es muy probable que en caso de producirse dicho evento, se precise reiniciar el PC del robot y si sigue sin establecerse comunicación habría que realizar un arranque en frío para reiniciar todas las variables de sistema o contactar con el servicio técnico para la verificación del estado del hardware del PC.

El evento será activado, si después de **10seg** de realizar la marcha de la instalación no se tiene la notificación de la señal de confirmación del robot de la Interface. Para este evento las señales implicadas serán las siguientes:

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

1. **%I5.1 -> Di_r_i_o_actconf** Entrada al PLC que llega desde el robot indicando que la Interface de comunicaciones está preparada.
2. **%MW118:X1 ->** Bit de alarma que activará el PLC para su aparición en la pantalla.

- **Alarma 3 (Accionamientos no preseparados).**



Los accionamientos del robot (**Ver capítulo 4 y 9**) También son notificados mediante evento, de esta manera sabremos si antes de proceder a la marcha del robot, hay que dar accionamientos, implicando una doble acción:

1. Habilitar seguridades (damos accionamientos)
2. Habilitar movimiento (damos marcha al robot).

Para dicha notificación se usará la señal empleada por el módulo automático externos, que informa sobre el estado de los accionamientos siempre y cuando no haya ninguna seguridad activada.

1. **%I5.0 -> Di_r_peri_rdy** Entrada al PLC que llega desde el robot indicando el estado de los accionamientos.
2. **%MW118:X2 ->** Bit de alarma que activará el PLC para su aparición en la pantalla.

3.3.5.7 Páginas de Sistema.

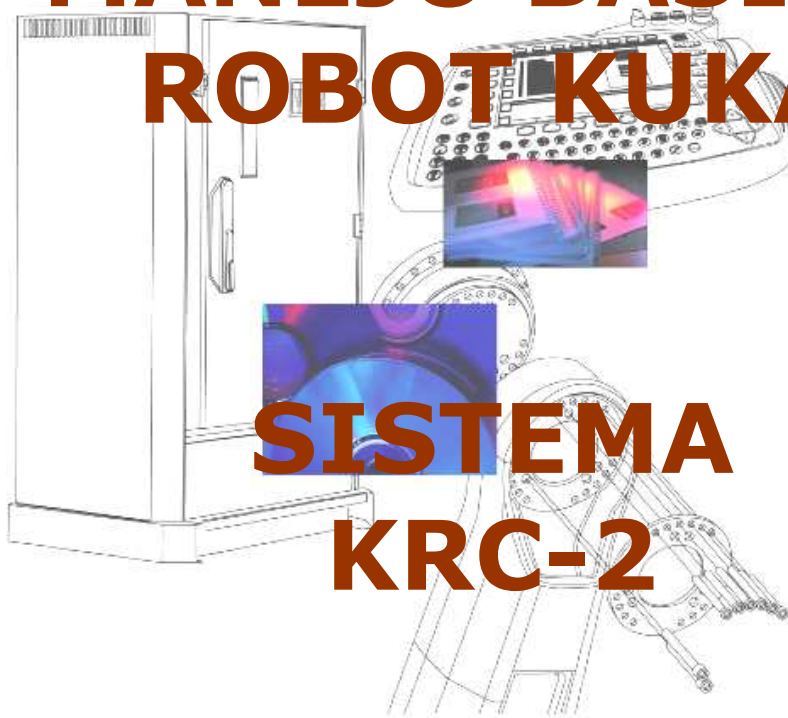
Las páginas de sistema, son páginas en forma de plantilla creadas por el diseñador del software, y cuyo formato gráfico (texto, imágenes u objetos estáticos) se aplica a otras páginas programas desde la aplicación que integran dicho formato.

Existen tres tipos de páginas de sistema.



1. Aplicación.
2. Alarma.
3. Ayuda.

Se encuentran disponibles con los terminales gráficos XBT, y pueden asociarse a las páginas de aplicación y de alarma. Las ventanas de ayuda pueden asociarse a cualquier campo de variable.

MANEJO BÁSICO ROBOT KUKA



SISTEMA KRC-2



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

4 SISTEMA ROBOT KUKA.

4.1 OBJETIVO DEL DOCUMENTO.

El sistema de un robot KUKA presenta el documento descriptivo empleado en este proyecto tanto de los elementos integrantes del mismo y puesta en servicio, como la sintaxis de programación a través de la KCP de un robot industrial KUKA; de ahora en adelante, brazo robotizado KRC-2.

Este documento está desarrollado; y en este concepto, determina el uso que se ha de hacer de él, para llevar a cabo e implantar los sistemas necesarios para la puesta en marcha del sistema de robot paletizado **PIENSOS O´COUTO (Fonsagrada-LUGO)**, es por lo tanto por lo que única y exclusivamente se hará hincapié en el manejo y sintaxis de programación desarrollando aquellas tareas que son objetivo para la consecución de tal propósito.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

4.2 ELEMENTOS BASICOS DE UN SISTEMA ROBOT.

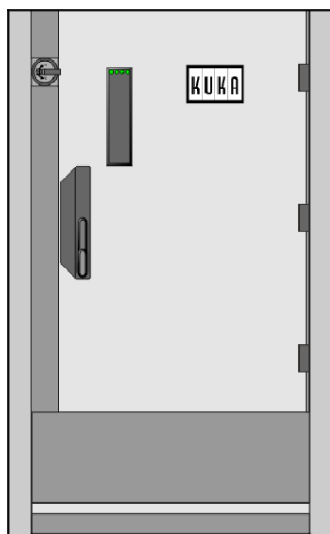
Los elementos básicos que integran un sistema KUKA (Fig.4.2-1); como cualquier otro tipo de sistemas robotizados; son el armario de control, el panel de programación o Panel control KUKA de ahora en adelante **(KCP)** y que se integra al PC de a bordo del armario, y por último el prototipo de brazo robot empleado que en nuestro caso será un modelo tipo **KR-180 P2** empleado para la paletización que soporta en su extremo 180Kg, junto con los cables de unión y accesorios.

(Fig.4.2-1) Componentes de un sistema completo Robot KUKA.



4.2.1 Armario de Control KR C2.

El armario de control es el elemento principal donde se integran todos los componentes que suministran potencia y controlan los servomecanismos del KR-180. Muestra las siguientes características técnicas:



(Fig.4.2-2) Armario KRC-2

Alimentación eléctrica: 3x400 V~ (-10%) a
3x415 V~ (+10%)

Frecuencia de red: 49 – 61 Hz

Potencia Nominal entrada según paquete:

-7.3 – 13.5 Kva.

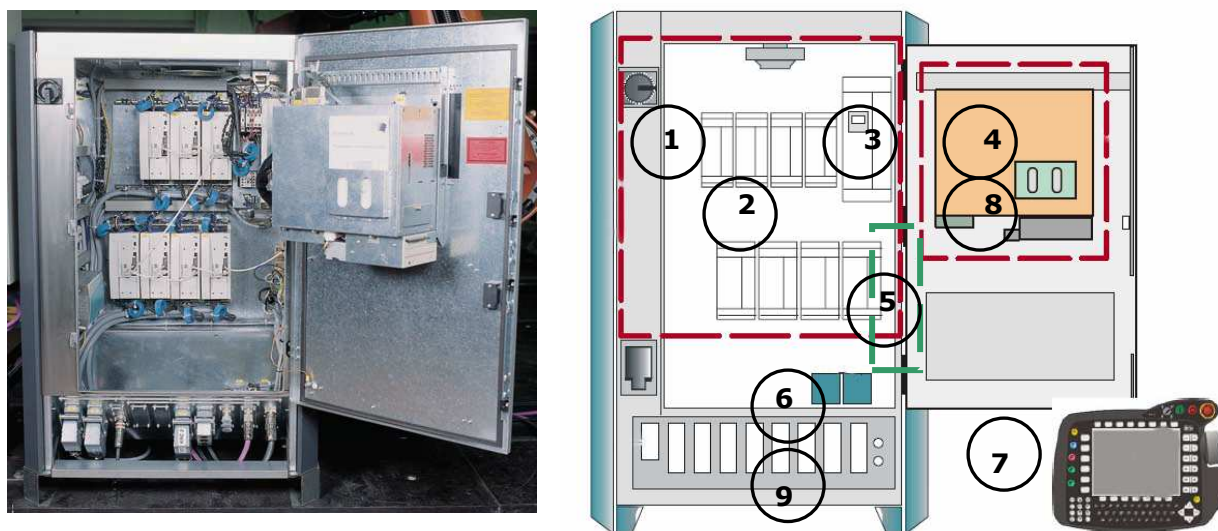
Clasificación de protección: IP 54.

Temperatura Ambiente:

-Sin unidad de refrigeración +5°C a 45°C
-Con unidad refrigeración +20°C a 55°C

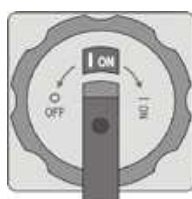
Peso Aproximado: 185 Kg.

Dentro de la KRC (Fig.4.2-3) se encuentran como elementos principales los servo convertidores o KSD (tantos como número de ejes presente el brazo robot) , la fuente de alimentación de potencia y baja tensión, protección de fusibles, tarjetas electrónicas (ESC de seguridades y multifunción MFC) ,refrigeradores, el PC de control con unidad de CD-ROM (para la instalación de software) y Floppy para la salvaguarda de seguridad de los programas, y unos acumuladores que suministran energía al PC durante algún tiempo transcurrido la falta de potencia.



(Fig.4.2-3) Elementos Armario KRC-2

1. Interruptor General.
2. Servo convertidores 4 a 6 ejes (**KSD_1 a KSD_6**).
3. Fuente de alimentación de potencia (**KPS-600**).
4. PC de control.
5. Tarjeta de seguridad (**ESC**).
6. Acumuladores.
7. Consola (**KPC**).
8. Tarjeta de Interface (**MFC**).
9. Conexiones Parte Inferior Conector de seguridades (**X11**).



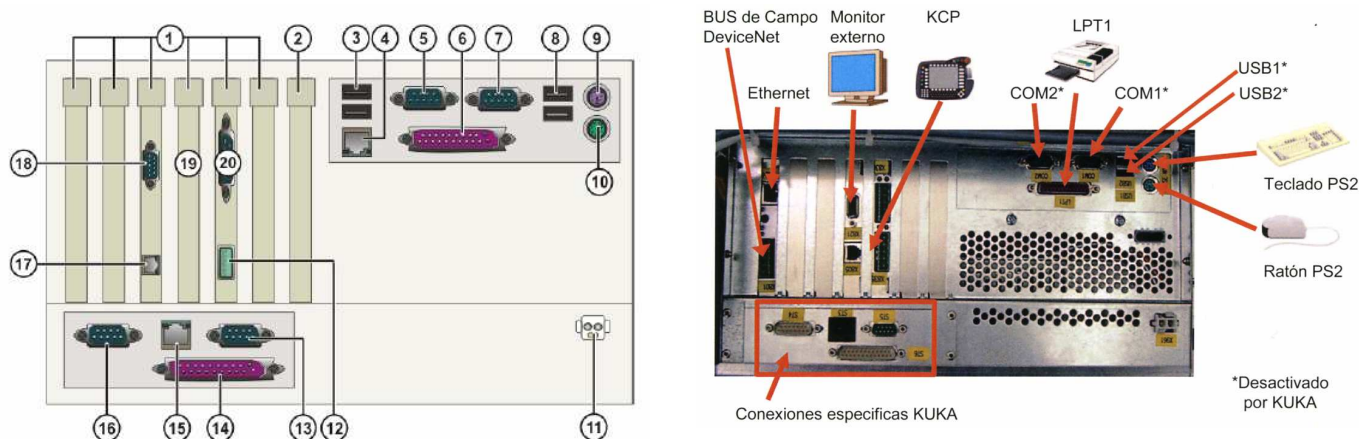
Con el interruptor General (1) se conecta el sistema robot, y es el único accionamiento no controlado. Así mismo en la desconexión del interruptor principal se cierra el software de KUKA y se apaga el sistema operativo transcurridos 2min de energía mantenida por los acumuladores.

En la parte inferior del armario se encuentran las conexiones; esto es, cables de alimentación/RED, cables de motor a robot, cables de señales del robot y conexión a KCP. El equipamiento estándar incluye además en el interior del armario, puerto Device Net y Ethernet para los sistemas de BUS habituales. Desde la KCP (7) y a través del ordenador de abordo (4), se maneja y opera la unidad de control del robot

4.2.1.1 PC de Control.

El PC de control opera bajo aplicación de Windows que esta destinado única y exclusivamente a la aplicación KRL (software de KUKA) y debe estar dimensionado con un mínimo de frecuencia de reloj de 733 MHz y una memoria RAM de 256 MB. A su vez incorpora las unidades de disco anteriormente mencionadas accesibles cuando la puerta del armario esta abierta. El resto del equipo es el Hardware estándar (fuente de alimentación, tarjeta gráfica KUKA con conexión a KCP). Existe la posibilidad de conectar un ratón estándar a través del puerto de comunicaciones COM1, así como un ratón/teclado con Interface PS/2. Se integra a modo de tarjeta PCI la tarjeta multifunción **MFC** con tarjeta digital **Servo DSE**. La MFC contiene el controlador de todas las entradas y salidas a modo de diferentes tipos de conexión **Device Net/CAN** formando la Interface entre la KCP y el ordenador, mientras que la DSE toma los valores nominales de los servos en cuanto a consumo y parametrización se refiere ; procesa por lo tanto, el estado de los servo controladores.

(Fig.4.2-4) Conexiones PC KRC-2





- | | |
|------------------------------|---|
| 1. Canales PCI. | 11. Alimentación 24V (Interface con tarjeta MFC). |
| 2. Enchufe AGP Pro. | 12. Connexion Interface Device Net (Input, Output). |
| 3. Puertos USB. | 13. COM 3 en tiempo Real. |
| 4. Conexión Ethernet. | 14. Conexiones ESC/KCP. |
| 5. Interface COM1 Serie. | 15. BUS de accionamientos KPS600. |
| 6. Interface LPT1 Paralelo. | 16. Interface RDW Serie. |
| 7. Interface COM2 Serie. | 17. Dysplay KCP (Tarjeta KVGA). |
| 8. Puertos USB. | 18. Conexión a Monitor a externo. |
| 9. Conexión de teclado PS/2. | 19. Conexión auxiliar tarjeta MFC. |
| 10. Conexión de ratón PS/2. | 20. In/Out tarjetas de seguridad. |

4.2.1.2 Unidad de alimentación KPS-600.

La **KUKA POWER Supply (Fig.4.2-5)**, suministra de energía a los contactores principales y alimenta los convertidores de servo. Mediante una alimentación rectificada a 27V c.c, se alimentan los frenos, reguladores del servo, se recargan los acumuladores y se alimentan las tarjetas electrónicas de seguridad.

La KPS está por lo tanto dividida en dos fuentes; la **KPS 600/20** (600V y 20Amp de potencia continua) y **KPS-27** (Baja tensión). Su Interface se encuentra conectada a la DSE. En la imagen adjunta se muestra las múltiples conexiones que integra.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

4.2.1.3 Acumuladores.

Estas baterías mantienen activa la memoria intermedia del sistema, de manera que los datos del ordenador se salven correctamente. Esto será siempre así, mientras la carga de los acumuladores no baje por debajo de los 19V, situación que será advertida mediante mensaje en el próximo arranque del equipo.

4.2.1.4 Servo convertidores KSD1.

Los servo convertidores (Fig.4.2-6) obtienen la etapa final rectificada para el control de cada uno de los ejes mediante regulación de corriente. Todos los KSD están conectados por interbus para la comunicación y conmutación entre ejes.



(Fig.4.2-6) Servoconvertidores (KSD).



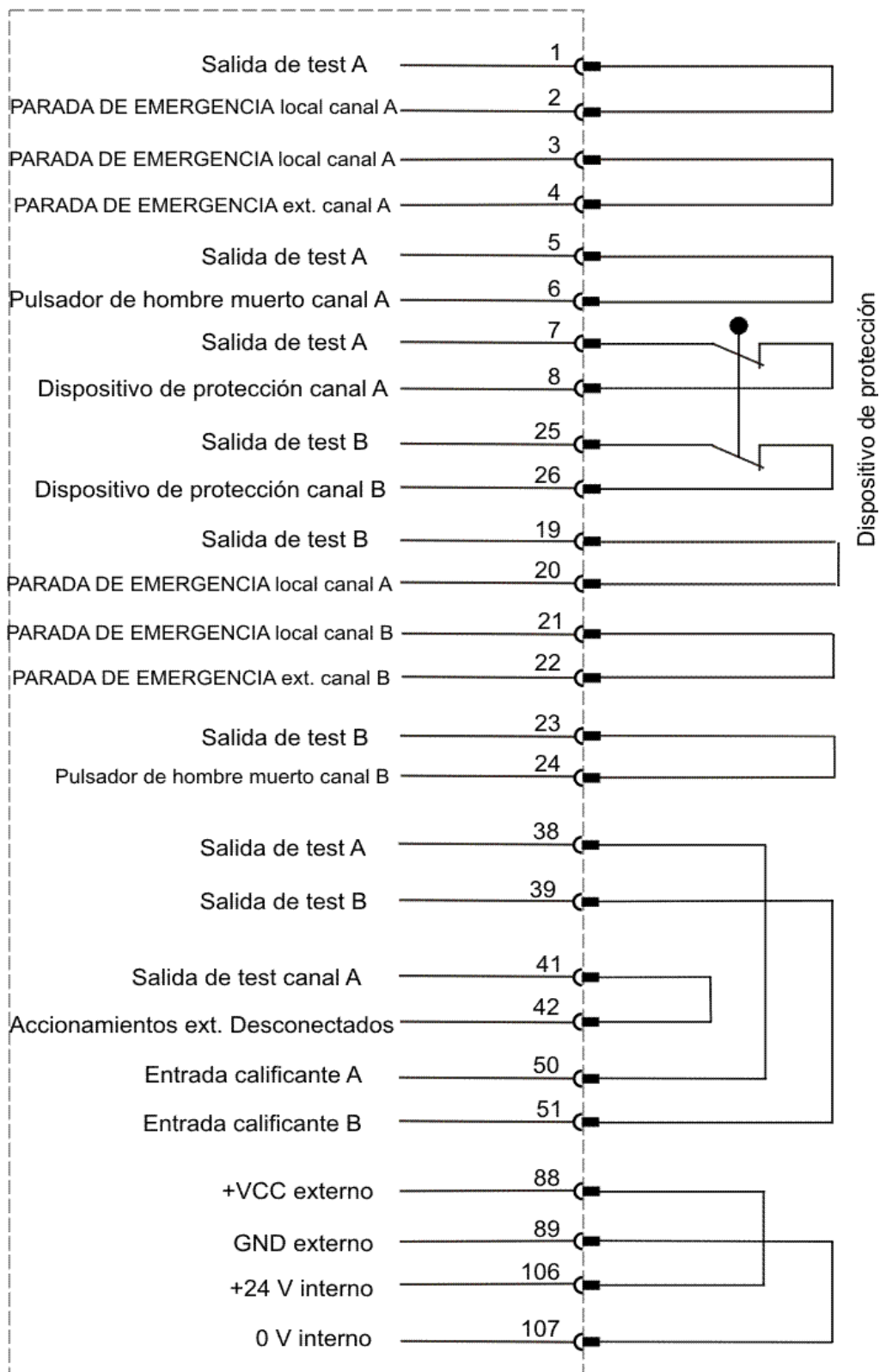
(Fig.4.2-5) KUKA Power Supply (KPS).

4.2.1.5 Conector (X11).

A través del conector de Interface X11, se conectan seguridades y setas de emergencia de la periferia concatenada con el sistema de KUKA. Dicho conector consta de una serie de pines desde lo que se interconectan los elementos de seguridad y se puntean aquellos que son necesarios para manejar el sistema.

En la Interface X11 se puede conectar la entrada de protección del operario, que sirve para el ínter bloqueo de dispositivos seccionadores de protección. A esta entrada bicanal pueden conectarse dispositivos de protección tales como puertas de protección. Si a esta entrada no se conecta nada, no es posible el modo de servicio automático. Para los modos de servicio T1 y T2 la protección del operario no se encuentra activa.

Dada su relevancia se presenta el siguiente diagrama de conexiones de dicha Interface, donde se muestran los canales de conexión en función del nivel de seguridad. Entre ellos se encuentran los canales A y B pines 7-8 y 25-26, que tendrán que ser seriados con la periferia para que los accionamientos se desactiven en caso de activarse alguna seguridad en modo automático. En la siguiente figura (Fig.4.2-7) se muestran el conector periférico y los puentes que son necesarios para la puesta en servicio en caso de no concatenar ninguna seguridad externa.



(Fig.4.2-7) Seguridades del conector X11.

4.2.2 KUKA Control Panel (KCP).

Es el elemento principal que hace de Interface con el ser humano, y con el programaremos y controlaremos al KRC-2, en ella se alojan todos los elementos de mando integrantes en un KRC. En el lado frontal encontramos las siguientes opciones de mando



(Fig.4.2-8) Kuka Control Panel (KCP).



(Fig.4.2-9) KCP Parte trasera.

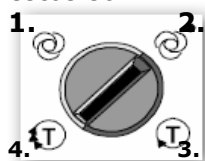
- | | |
|---------------------------------|--|
| 1. Display o Monitor en color. | 10. Teclado numérico. |
| 2. Accionamientos en ON. | 11. Teclas de función. |
| 3. Accionamientos en OFF. | 12. Tecla de arranque hacia atrás. |
| 4. Pulsador Seta de emergencia. | 13. Tecla de arranque. |
| 5. Mouse Espacial 6D. | 14. Tecla de Stop. |
| 6. Teclas de estado derecha. | 15. Tecla de cambio de ventana. |
| 7. Tecla de entrada. | 16. Tecla de escape ESC. |
| 8. Cursor. | 17. Teclas de estado izquierda (menú). |
| 9. Teclado Alfanumérico. | 18. Teclas de menú. |

- | |
|--|
| 1. Placa de características.
2. Tecla de Arranque.
3. 4 y 5 Pulsador de Hombre muerto con tres posiciones. |
|--|

En el aparatado de accionamientos y empezando de izquierda a derecha, tenemos en primer lugar el selector de modos de servicio, desconexión de los accionamientos, conexión de los accionamientos y seta de emergencia.



4.2.2.1 Selector de Modos de servicio (1).

Conmuta entre los cuatro modos de funcionamiento establecidos en el KRC, estos son:



- 1.-Automático.
- 2.-Automático Externo o controlado por PLC.
- 3.-Manual (Test 1).
- 4.-Manual (Test 2).

En el modo automático, el arranque de ejecución de un programa se ejecuta mediante selección del mismo en la consola KCP y pulsador de marcha o tecla de

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

arranque a la velocidad programada, mientras que en el modo automático externo la selección y marcha de un programa se ejecuta por accionamientos externos como pudiera ser un PLC.

La selección de modos en manual Test (T1 o T2) se ejecutan movimientos en manual de los ejes o movimientos pasa a paso de un programa manteniendo pulsada la tecla de hombre muerto y marcha a la vez, pero mientras en T1 se ejecuta el movimiento a la velocidad configurada en manual (máximo 25% de la velocidad en automático), en T2 se ejecutan movimientos de programa a la velocidad de automático.



El servicio en modo automático sólo es posible si el circuito de seguridad ESC esta cerrado.

4.2.2.2 Accionamiento en ON/OFF (2)(3).



En condiciones normales del servicio, acciona los accionamientos, desencavando los servomecanismos para su posterior movimiento. En definitiva se preparan los motores para ser accionados.



Activan con un breve retardo los frenos de los motores que mantienen los ejes en su posición, desconectando así, los accionamientos.

4.2.2.3 Pulsador Seta de emergencia (4).



El pulsador de PARADA DE EMERGENCIA del sistema del robot se encuentra sobre el KCP. Cuando se pulsa la PARADA DE EMERGENCIA, en los modos de servicio T1 y T2 se desconectan los accionamientos de forma inmediata.

El pulsador de PARADA DE EMERGENCIA debe pulsarse siempre cuando peligran personas o partes de la instalación.

4.2.2.4 Teclas de Arranque y parada (12), (13) y (14).



Mediante la tecla de arranque (+) se inicia la ejecución del programa en los modos de servicio automático; siempre y cuando los accionamientos estén conectados. En los modos T1 o T2 se ejecuta el programa hacia delante paso; siempre y cuando se mantengan pulsada la tecla a la vez que el hombre muerto, mientras que el arranque (-) procesa la ejecución pasa a paso del programa en sentido contrario. La tecla de Stop o parada del programa, detiene el puntero de programa en modo automático, provocando una parada sobre la trayectoria.

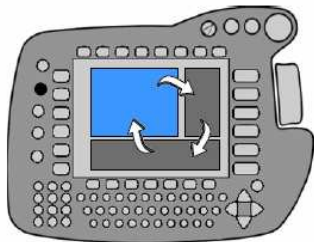
4.2.2.5 Tecla de Validación (7) y cambio de ventana (15).



Confirmación o validación de datos así como de las funciones seleccionadas.



Permite conmutar entre la ventana de programa, la ventana de estados y la ventana de mensajes, quedando resaltada la ventana activa de color azul.



(Fig.4.2-10) Cambio de ventana



(Fig.4.2-10) Cambio de estados

4.2.2.6 Teclas de estado derecha (6), (17), (18) y de función (11).

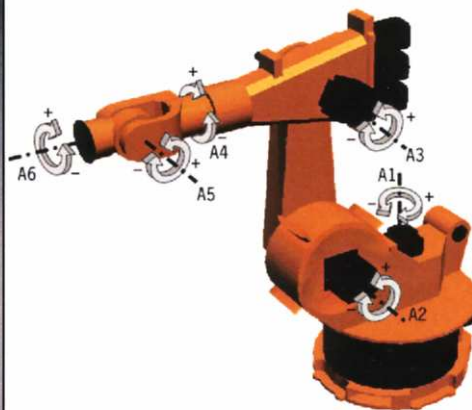
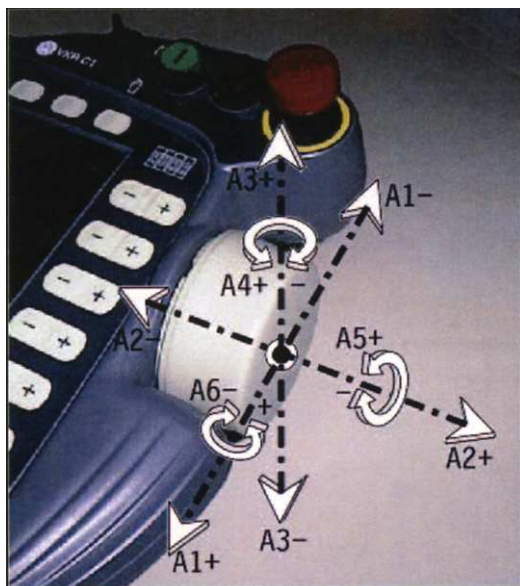


Seleccionaremos diferentes opciones dependiendo del modo de servicio en el que nos encontremos, para conmutar funciones y definir valores. Las teclas de función variaran su significado en función de lo que se nos vaya mostrando en pantallas y las teclas de menú permiten abrir el menú correspondiente (Fig.4.2-11).

4.2.2.7 Space-Mouse (Ratón espacial) (5).

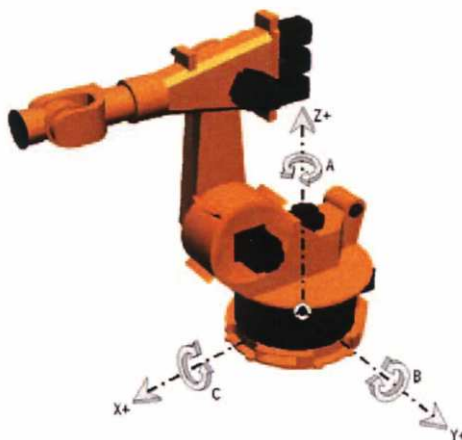
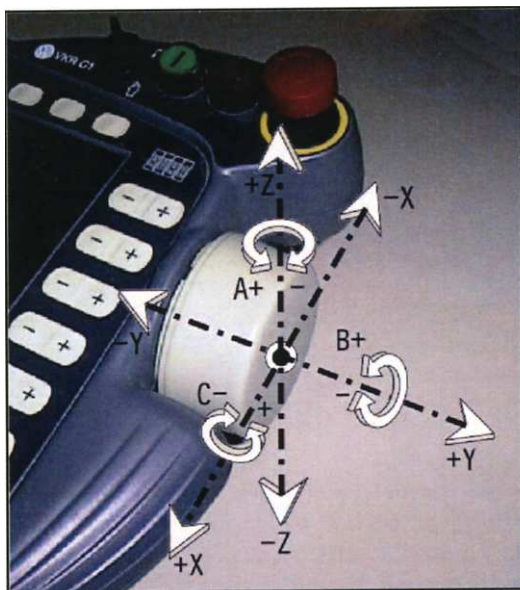


Se controlan manualmente los 6 ejes o grados de libertad del robot. Así vemos en las siguientes imágenes los movimientos que podemos desarrollar con dicho mando tanto para movimientos eje a eje como cartesianos.



(Fig.4.2-11) Movimiento SPACE-MOUSE eje a eje.

En los movimientos eje a eje (Fig.4.2-11) vemos el ángulo de giro positivo o decreciente de los ejes A1, A2, A3, A4, A5 y A6. Mientras que en movimientos cartesianos (Fig.4.2-12) podremos mover en X, Y, Z con las respectivas orientación de la herramienta A,B,C.



(Fig.4.2-12) Movimiento SPACE-MOUSE Cartesiano con control de la Orientación

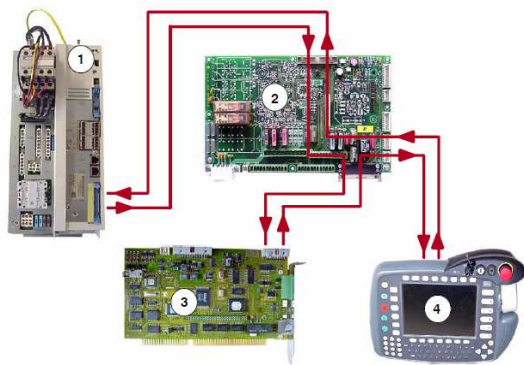
4.2.2.8 Pulsador de Hombre Muerto (3) Parte Trasera..

Los pulsadores de hombre muerto de tres posiciones, dejan conectar los accionamientos en los modos de servicio T1 y T2.

En los modos de test, el robot sólo puede ser desplazado cuando el pulsador de hombre muerto se encuentra en la posición central. Al soltar o pulsar completamente (posición de pánico) el pulsador de hombre muerto, los accionamientos se desconectan inmediatamente provocando la parada del robot. En los modos de servicio T1 y T2, el pulsador de hombre muerto debe mantenerse **en la posición intermedia** para poder efectuar los movimientos en manual. En los modos de servicio Automático y Automático Externo, el pulsador de hombre muerto carece de función.



4.2.3 Tarjeta electrónica (ESC).

Esta tarjeta se utiliza como bus de seguridad para interconectar todos los elementos integrados ante un mal funcionamiento del sistema de control o una emergencia de la periferia externa. Será el dispositivo mediante microcontroladores que haga de Interface con todo el sistema de control.



(Fig.4.2-13) Interconexión BUS de seguridad.

Los componentes que interconectan son la fuente de alimentación de potencia o **KPS**, la consola de control o **KCP** y la tarjeta de control de entradas y salidas **MFC**, siendo los circuitos eléctricos destinados a la seguridad los dispositivos de parada de emergencia, pulsador de hombre muerto, protecciones de usuario y modos de servicio, ya que en función del sistema de seguridad que actúe, la reacción de la ESC será de un tipo u otro.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

1. Elemento KPS.
2. ESC.
3. MFC (Elemento Pasivo seguridades controladas por periferia).
4. KPC o elemento MASTER de seguridad.

De esta manera cualquier mal funcionamiento de uno de los componentes es reconocido y llevado a un estado seguro de operación, a la vez que se desconectan la alimentación de los accionamientos en la KPS provocando una parada cinemática de los ejes.

El elemento integrante KCP ejerce frente al sistema de seguridad de la ESC, de MASTER; es por lo tanto, desde donde se inicializa el circuito de protección a través del accionamiento seta de emergencia, pulsador de hombre muerto y otras funciones como el modo de operación explicado anteriormente.



Es preciso que la KCP; al integrar una seguridad elemental como es la seta de emergencia, se encuentre conectada al armario de control para cerrar el circuito ESC; en caso contrario la desconexión de los accionamientos se produce de manera inmediata sin ningún tipo de retardo.

La ESC; para asegurar la integridad del sistema, ejecuta constantemente un análisis de fallos internos que provoquen la desconexión del KRC, como pueden ser cortocircuitos a través de los canales de seguridad en módulos PILZ, errores de comunicación y Hardware. Siendo el retardo para la desconexión en caso de parada de emergencia de 0.7 a 1.5 seg.

4.3 ARRANQUE DE LA UNIDAD DE CONTROL.

Una vez iniciado el arranque se muestra en la consola el menú básico que permite seleccionar y ejecutar programas. Básicamente existen dos métodos diferenciados de arranque. **Arranque en caliente y Arranque en frío.**

En el arranque en caliente se reestablecen las señales de la instalación al estado tal cual habían quedado antes de la falta de energía. Se establecen las salidas y se inicia la Interface con la tarjeta MFC, pudiendo volver a trabajar desde el punto coordenada o trayectoria desde el que se había quedado el sistema.

El arranque en frío se produce antes del inicio de ejecución del programa seleccionado. Dicho arranque puede ser llamado en cualquier momento mediante las opciones de configuración del software, para que en el próximo arranque se reestablezca todo el sistema por completo.

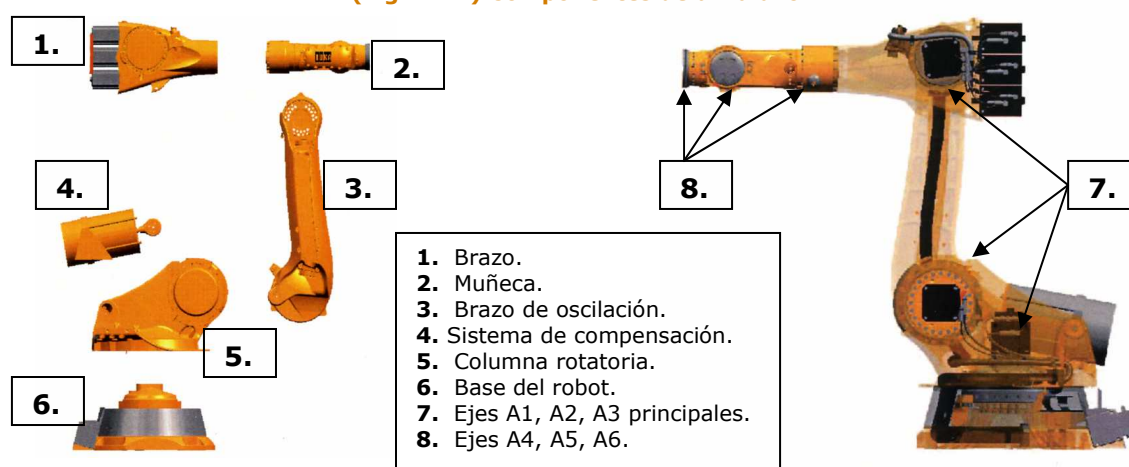
En caso de corte de suministro temporal, si este es reestablecido antes del acuse power-off de los acumuladores, el equipo no se apaga y procederemos a operar confirmando según que indicaciones en pantalla. Si el servicio no es reestablecido, el equipo se apaga después del power-off de manera controlada hasta el siguiente arranque. No obstante; y dependiendo de si el robot estaba ejecutando una trayectoria en el momento del corte, pudiera ser que en el próximo arranque hubiera que forzar una coincidencia de trayectoria o COI ALCANZADA (como veremos en próximos apartados) para proseguir con el próximo movimiento en ejecución del programa. Siempre que se produce un corte en el suministro, los datos se salvan automáticamente mediante los acumuladores instalados en el armario de control.

4.4 SISTEMA DE COORDENADAS.

Para poder indicar en el espacio la posición u orientación de un punto, se utilizan diferentes sistemas de coordenadas. Estos sistemas de coordenadas determinan los movimientos en un brazo robot KUKA. Se pueden realizar cuatro tipos de movimientos, **Eje a Eje**, **WORLD**, **TOOL** y **BASE**. De estos cuatro sistemas de movimiento los más empleados en manual son los movimientos específicos de ejes y el sistema de coordenadas World, ya que los otros dos son generados por el usuario.

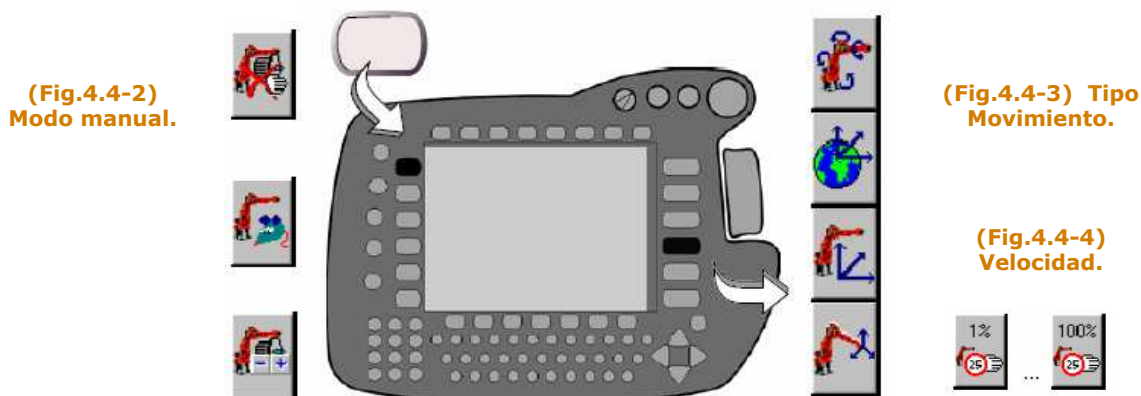
Para comprender mejor como se comporta un brazo robot eje a eje, haremos una breve descripción y desglose mecánico, formándolo las siguientes partes elementales:

(Fig.4.4-1) Componentes de un brazo



4.4.1 Movimientos Manuales.

Estos movimientos son habilitados mediante el selector de movimientos (Posición T1, T2), y posteriormente mediante las teclas de modo seleccionaremos si queremos realizar los desplazamientos mediante el ratón 6D o el teclado. También tendremos la opción de seleccionar la velocidad de movimientos en manual (10%-25% sobre la velocidad real) y una vez accionados el pulsador de Hombre muerto (donde los accionamientos en la barra de estado pasarán a color verde) ejecutaremos el movimiento a realizar.



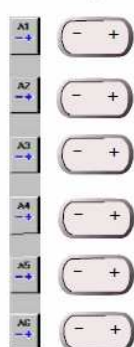
4.4.1.1 Sistema de coordenadas específico eje a eje.



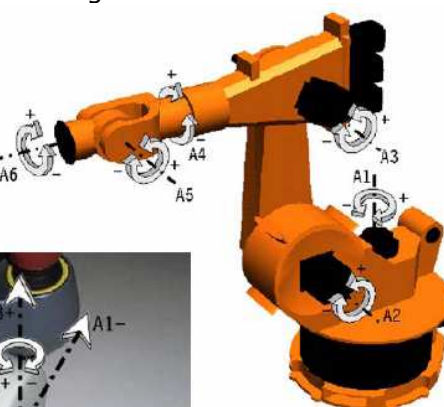
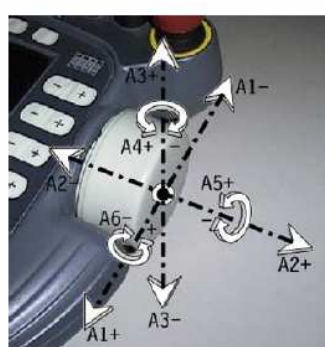
- **Movimiento específico de ejes.**

Cada eje del robot puede moverse individualmente en dirección AXIAL positivo o negativo según la siguiente imagen.

(Fig.4.4-5) Teclas de desplazamiento.



(Fig.4.4-6) Ratón 6D.



(Fig.4.4-7) Movimiento específico eje a eje.

En la barra de teclas a la derecha se indican los ejes a seleccionar, o bien el ratón 6D, de esta manera una vez que no haya ninguna seguridad activa y con el pulsador de hombre muerto, activaremos los accionamientos en ON. Para el caso de un robot de 6 ejes, seleccionaremos el eje a mover **A1, A2, A3, A4, A5 ó A6** correspondiente, donde los ejes A1, A2 y A3 son los principales, A1 la base del robot y A6 el eje final de la herramienta del robot. Para el caso del robot utilizado en la aplicación, tendremos 4 ejes; siendo estos, **A1, A2, A3, A6** donde A4 y A5 no serán utilizados. Estas coordenadas se guardan en variables AXIS de tipo estructura que ya se encuentran predefinidas en el sistema.

- A1.** Sentido horario (+) con área de trabajo desde +185° hasta -185°.
- A2.** Sentido antihorario (+) con área de trabajo desde +0° hasta -129°.
- A3.** Sentido antihorario (+) con área de trabajo desde +161° hasta -19°.
- A4.** Sentido horario (+) con área de trabajo desde +350° hasta -350°.
- A5.** Sentido antihorario (+) con área de trabajo desde +118° hasta -118°.
- A6.** Sentido horario (+) con área de trabajo desde +350° hasta -350°.

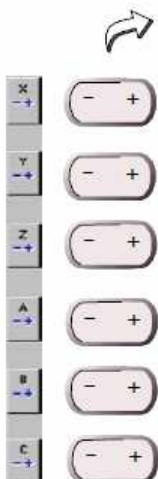
4.4.1.2 Sistema de coordenadas en WORLD.



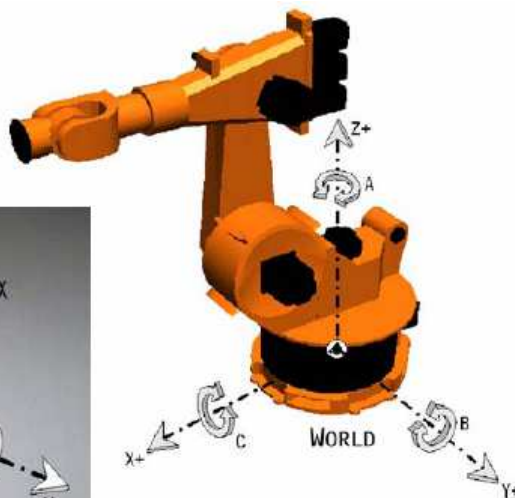
- **Sistema de coordenadas WORLD.**

Es un sistema de coordenadas cartesiano cuyo origen se encuentra en la base del robot. Mientras que el sistema de coordenadas **BASE** el origen de se encuentra en una pieza y el sistema **TOOL** sobre la herramienta.

(Fig.4.4-8) Teclas de desplazamiento.



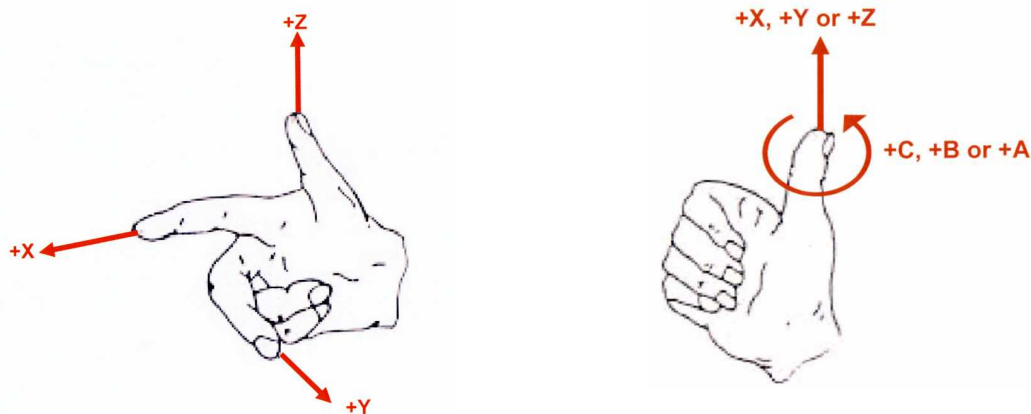
(Fig.4.4-9) Ratón 6D.



(Fig.4.4-10) Sistemas de coordenadas

En dicho sistema seleccionaremos con respecto a la base del robot, si este será desplazado en **X**, **Y** o **Z**, además le indicaremos la orientación **A**, **B** o **C** del eje final donde se ancla la herramienta, donde **A** será el giro sobre la coordenada **Z**, **B** el giro sobre la coordenada **Y**, y **C** el ángulo de rotación sobre **X**. El sistema de ejes se desplazará sincrónicamente para provocar el desplazamiento a lo largo de las coordenadas.

Para definir el sentido positivo o negativo de un movimiento, debemos obedecer a la regla básica de la mano derecha (Fig.4.4-11) donde se muestra la dirección del sistema de coordenadas y del sistema de rotaciones tal como sigue.



(Fig.4.4-11) Regla de la mano derecha

De esta manera y situados en la parte trasera de robot, se definen los sentidos positivos en los movimientos cartesianos, así como la dirección de las rotaciones.

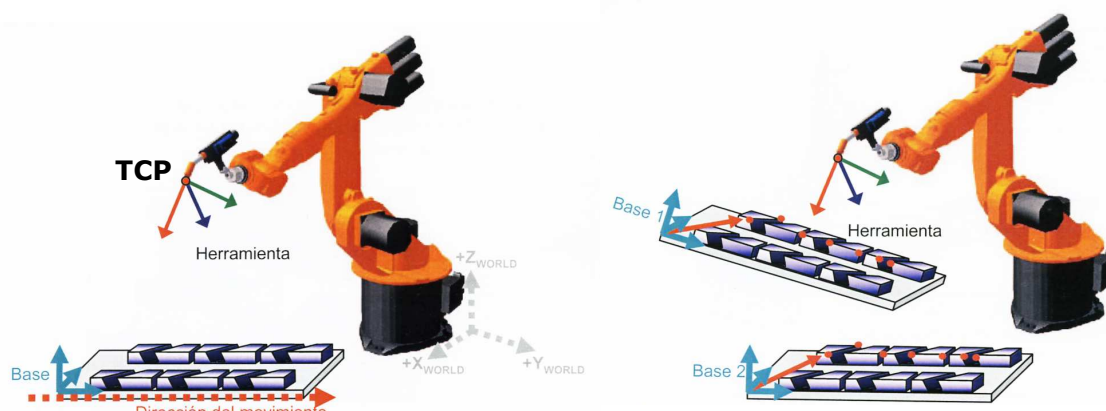
La dirección es configurable desde la **KCP**, donde a través del menú **Configuración** podremos configurar; entre otros, la posición del puntero y del ratón 6D para cambiar la dirección positiva de los movimientos.

4.4.1.3 Orientación de la herramienta STATUS "S" TURN "T".

El desplazamiento punto a punto implica que un mismo punto en el espacio, puede ser alcanzado con diferentes posiciones angulares de los ejes, de manera que para diferentes valores en el espacio de **X, Y, Z** se obtiene el mismo punto cambiando la posición mecánica de la herramienta. Para conseguir que una posición sea unívoca existe otra característica a tener en cuenta que es la orientación de la herramienta que se define mediante las variables **"S" (Status)** y **"T" (Turn)** y determinan la orientación de la muñeca. Estos valores solamente se evalúan en los movimientos **PTP**. Por esta razón, el primer movimiento en un programa debe ser un movimiento de **PTP**.

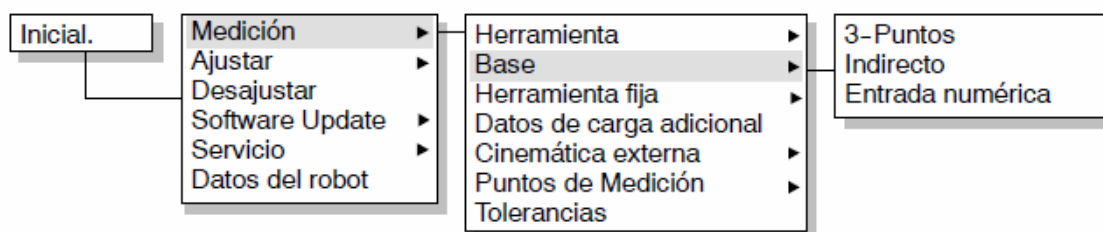
4.4.2 Medición de una base.

Para una instalación de robot paletizado es muy común el uso de bases, tantas como espacios de palatizados existan. Esto es debido a que podemos cambiar el sistema de coordenadas cartesianas en base al robot para que se muevan según la posición del palet, de manera que podamos modificar los puntos siempre paralelo y perpendicular al palet. Por lo tanto nuestra área de trabajo recibirá un sistema de coordenadas cuyo origen será definido por nosotros mismos, y cuyos puntos se incrementarán con respecto a las direcciones de esa base.



(Fig.4.4-12) Medición de una base.

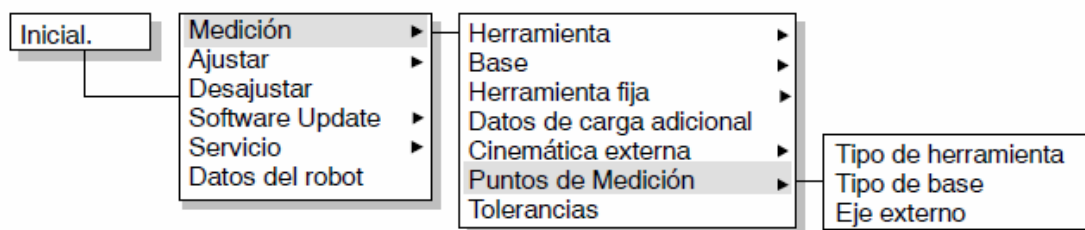
Existen varios métodos para crear bases, de entre ellos el más extendido es el llamado **Método de los tres puntos**. Desde el menú **Inicial** se seleccionará dicho método.



En dicho método se posicionará manualmente el robot al punto de origen, dirección de desplazamiento en X y punto del plano que forma la nueva base. Se guardan las nuevas coordenadas y la nueva base que podremos referenciar a los puntos de trayectoria del programa. Los pasos a seguir serán los siguientes:

1. EL **TCP** (punto de la herramienta a implementar) se mueve al origen del nuevo sistema de coordenadas. Aceptaremos el nuevo valor mediante la tecla programable **OK**.
2. El **TCP** de referencia es movido hacia el punto **X** positivo del nuevo sistema.
3. En el Tercer paso el punto **TCP** es movido a un lugar del plano **X-Y** de manera que **Z** se calibrará automáticamente.

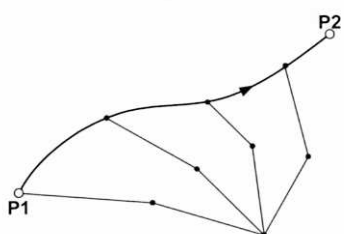
Bajo el siguiente menú, se pueden visualizar los datos de medición guardados.



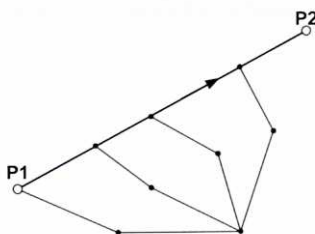
4.5 MOVIMIENTOS DEL ROBOT.

En un sistema de robot, podemos observar tres tipos de movimientos:

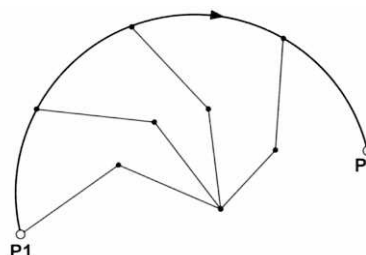
1. **PTP.** Donde la herramienta **TCP** se desplaza de un punto a otro con la trayectoria más rápida (Fig.4.5-1). Este tipo de movimiento es el más rápido y cómodo para los ejes del robot.
2. **LIN.** La herramienta se desplaza de un punto a otro mediante una línea recta a una velocidad definida (Fig.4.5-2). El robot debe corregir constantemente la trayectoria para mantener la linealidad, por lo que la velocidad es más lenta.
3. **CIRC.** La herramienta es guiada de un punto a otro a través de una trayectoria circular (Fig.4.5-3).





(Fig.4.5-1) Movimiento PTP.



(Fig.4.5-2) Movimiento LIN.



(Fig.4.5-3) Movimiento CIR.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

4.5.1 Punto de trayectoria alcanzada (COI).

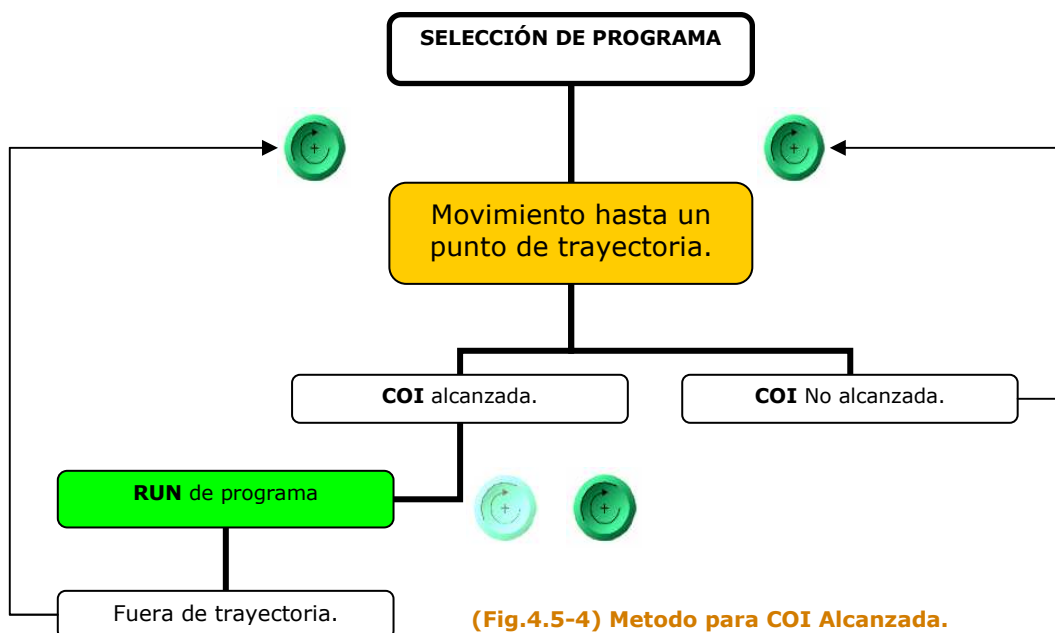
Para garantizar una concordancia de la posición del robot con las coordenadas del punto programado, se ejecuta el llamado desplazamiento COI (desplazamiento de coincidencia). Este desplazamiento se efectúa a velocidad reducida, y es necesario para asegurar que el robot siempre se encuentra en una trayectoria determinada por los puntos que están programados, y así tener la seguridad que nos encontramos dentro de la zona de trabajo establecida. Será necesario ejecutarlo en las siguientes circunstancias:

- Después de devolver un programa a su estado inicial mediante un desplazamiento COI hasta la posición de partida.
- Después de una selección de paso sobre las coordenadas del punto en el que se encuentra el puntero de paso.
- Después de seleccionar el programa "CELL" antes de poder arrancar el modo de servicio en automático externo.
- Después de haber seleccionado un nuevo programa.
- Después de modificar una instrucción.
- Después de un desplazamiento manual durante el servicio con programa
- Cuando el punto de referencia de la herramienta se encuentra fuera de la ventana de posicionamiento.

El punto denominado **HOME** suele ser aquel en el que iniciamos el programa, y se suele poner al principio y final ya que representa una posición definida de seguridad en la ejecución. Será por lo tanto el punto desde donde arrancaremos el programa.

La manera de realizar el **COI (Fig.4.5-4)** de trayectoria es manteniendo apretada la tecla **RUN** (Avance hacia delante) en el arranque del programa, de manera que el robot se moverá a velocidad reducida hasta que se alcance un punto de trayectoria programado. Una vez alcanzado el punto de trayectoria (indicado mediante mensaje) soltaremos la tecla de arranque y volveremos a presionarla para proseguir con la ejecución ya en automático.

En el programa de paletizado que nos ocupa, suele ser habitual la programación tanto al inicio y final de un subprograma de coincidencias de trayectorias en función de la posición actual del eje A1. Este subprograma sería **coi_sgrd ()** y su sintaxis se encuentra en los anexos del programa, donde en función de la posición actual (en grados de giro) del eje A1, se ejecutan una serie de movimientos a unos puntos definidos fuera de cualquier tipo de obstáculo en el recinto antes de posicionarse en el punto **HOME**.



```

;=====ENCIMA MESA RECOGIDA=====
if (($axis_act.a1>=-50) and ($axis_act.a1<=-35)) then
PTP HOME CONT Vel= 100 % PDAT17
endif

```

PTP HOME CONT Vel= 100 % PDAT19

Ejemplo Programa
Coi_sgrd ()

En esta parte del programa si el eje principal **A1** se encuentra entre -35° y -50°, significará que estará encima de la zona de los rodillos de alimentación o mesa de recogida. En este se realizará un movimiento al pto **HOME** directamente.

Cualesquiera de los puntos de programa **P1,P2,P3,P4,P5,P6,P7,P8 y HOME**, serían puntos de **COI ALCANZADA** (Ver ANEXO Coi_sgrd()).

4.5.2 Movimiento (PTP).

Para ejecutar una trayectoria desde un punto **PTP** a otro **PTP**, todos los ejes se mueven de manera sincronizada, de forma que el movimiento ejecutado es el más cómodo y rápido para los ejes del robot. Independientemente de la velocidad que adquiera cada uno de los ejes, todos ellos finalizan en dicho punto.

4.5.2.1 Programación de un movimiento PTP.

Si seleccionamos la opción **Movimiento->PTP** del menú, se nos desplegará en pantalla el formulario que nos permitirán definir las características de dicho movimiento. Estas son las siguientes:

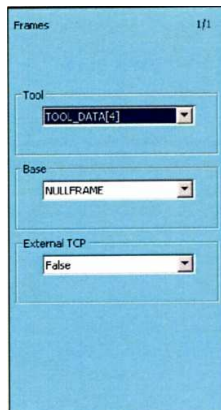
PTP P1 CONT Vel=100 % PDAT1

CONT

- **PTP** -> Aquí se seleccionará el tipo de movimiento.

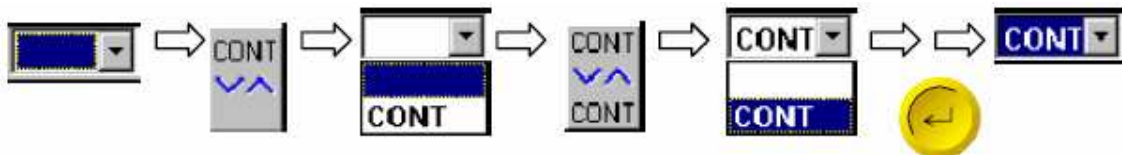


- **P1** -> Es el nombre que le damos al punto P1, P2....HOME (Máximo 23 caracteres). En la ventana de estados nos aparecerán las siguientes opciones:



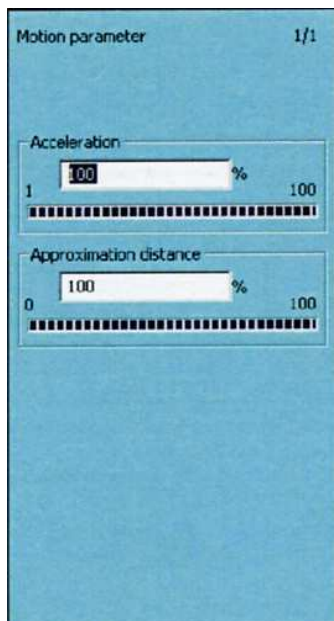
1. Tool o herramienta seleccionada (hasta 16 herramientas); que en nuestro caso será Nula o la misma para todos los puntos **Nullframe**.
2. Base a la que está referenciado dicho punto.
3. TCP Externo en la que se muestra **false o true** si el robot conduce o no la herramienta.

- **CONT** -> Seleccionamos si queremos ir a las coordenadas exactas del punto o si queremos ir con una aproximación.



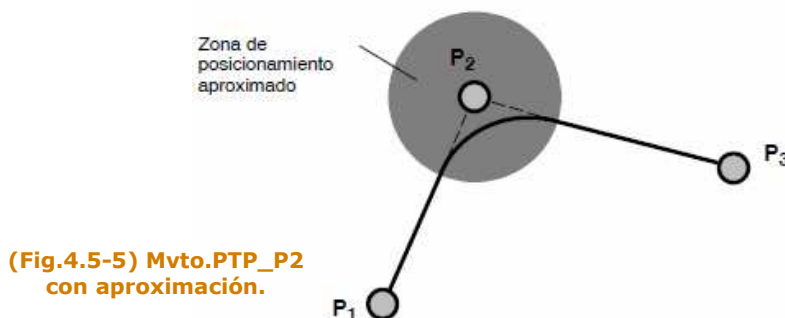
Durante la aproximación se consigue que el robot no se frene totalmente en cada punto, con lo que la velocidad del ciclo mejora y se reduce el desgaste, pero es importante tener en cuenta que la trayectoria a seguir por el robot no es fiable y previsible.

- **Vel** -> Velocidad con la que queremos ir al punto.
- **PDAT1** -> Son los parámetros de movimiento, En dicho campo se nos mostrará las opciones donde se define la aceleración y distancia de aproximación.



Aceleración: Rango de valor de 1% a 100%

Aproximación: Rango de valor de 0% a 100% siendo el 100% el valor máximo equivalente a 300mm de aproximación.



Una vez posicionado el robot en el lugar donde queremos incorporar el punto y rellenado los campos del formulario, el sistema nos preguntará si queremos utilizar las coordenadas actuales para ser salvadas en el punto en edición, a lo que contestaremos que sí.

Un movimiento PTP se puede programar sin emplear la plantilla o herramienta que proporciona KUKA por defecto. Para la programación mediante sintaxis se hace uso de variables predefinidas del sistema de tipo estructura, cuyos datos son cada uno de los valores de los ejes del robot o las posiciones cartesianas. La forma de declarar un punto mediante línea de comandos sería la siguiente:

PTP {AXIS: A1 0,A2 -90,A3 90,A4 0,A5 0,A6 0}
PTP {POS:X 1025,Y 0,Z 1480,A 0,B 90,C 0,S 'B 010',T 'B 000010'}

En esta instrucción estamos declarando un punto con las coordenadas angulares A1...A6 correspondientes y mediante las coordenadas cartesianas.

PTP {A3 45}
PTP {Z 1300, B 180}

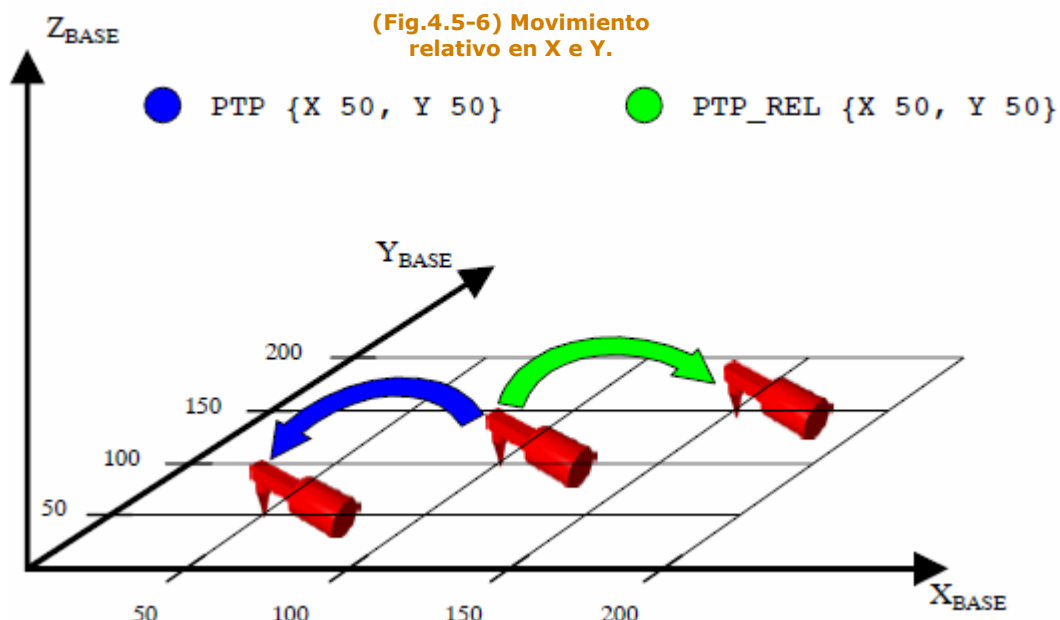
Mediante esta instrucción desplazamos el eje A3 a su posición angular 45° o nos movemos en elevación 1300mm con orientación B de la herramienta de 180°

También es muy común realizar movimientos relativos con respecto a la posición actual, así por ejemplo la siguiente instrucción desplazaría los ejes A1 y A4 15° sobre su posición actual, o subiríamos en Z 180mm con respecto al valor actual con orientación TCP -90°.

PTP_REL {A1 15, A4 15}.
PTP_REL {Z 180, B -90}

Añadiendo la palabra clave **"C_PTP"** estaremos indicando que queremos realizar el movimiento con contorno.

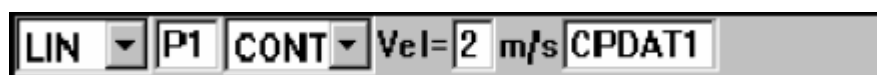
PTP {POS:X 1246.93,Y -98.86,Z 715,A 125.1,B 56.75,C 111.66,S 2,T10}
C_PTP



4.5.2.2 Programación de un movimiento LIN.

Los movimientos lineales son empleados para aproximaciones a velocidad reducida y constante, de manera que podamos acercarnos con mayor exactitud al punto objetivo en concreto, en este tipo de movimiento el estado del punto final es igual que el inicial por lo que no son necesarios el uso de variables **s,t**.

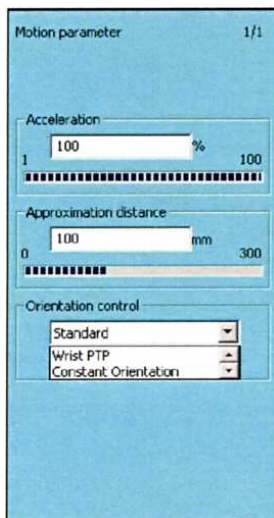
En dicho caso la declaración de un punto **LIN** es exactamente la misma que para un **PTP** con alguna diferencia en concreto.



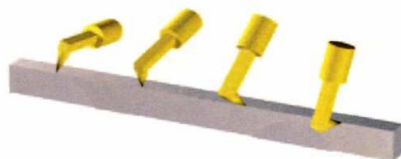
En este caso la velocidad máxima que se podrá alcanzar será de 2m/seg. siendo la mínima de hasta 0,001m/seg. Dichos movimientos admiten la parada exacta o aproximada con la función de contorno.

En la lista de parámetros **CPDAT1** se añaden a parte de la aceleración y la distancia de aproximación, el control de la orientación que puede ser de tres tipos:

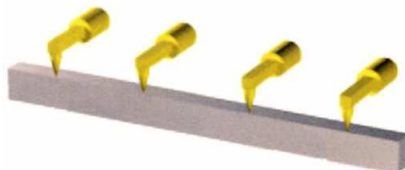
1. **Estándar.** Donde la orientación de la herramienta cambia de manera continua de inicio a fin mediante giro y rotación de la herramienta.
2. **PTP Manual.** Se define la orientación continua de los ejes para evitar singularidades de giro en los ejes del robot (ángulos imposibles).
3. **Constante.** Se mantiene la orientación de la muñeca.



(Fig.4.5-7) Control de la orientación standard.



(Fig.4.5-8) Control de la orientación constante.



La sintaxis de programación para estos tipos de movimientos es similar que en el caso anterior, siendo "LIN" o "LIN_REL".

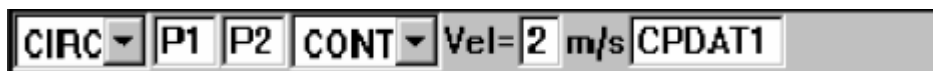
LIN {X 1030,Y 350,Z 1300,A 160,B 45,C 130}

LIN {POS: Y 0,Z 800,A 0,S 2,T 35} S,T se ignoran en este caso.

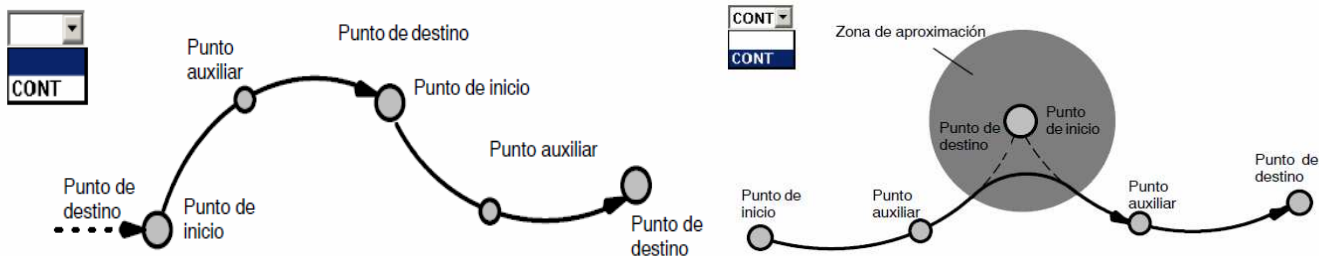
LIN_REL {FRAME: X 300} Movimiento relativo lineal a lo largo del eje x.

4.5.2.3 Programación de un movimiento CIRC.

Mediante este tipo de movimientos, conseguimos mover el TCP de un punto a otro mediante una trayectoria circular. Para definir dicha trayectoria es necesario declarar un punto auxiliar para calcular la circunferencia hasta el punto final. La declaración sigue las mismas pautas que en los movimientos vistos anteriormente, con la inclusión de un nuevo campo para la definición del punto auxiliar.



- **P1.** Será la definición del punto auxiliar declarado anteriormente sobre el plano de giro (máximo 23 caracteres).
- **P2.** Será la definición del punto final

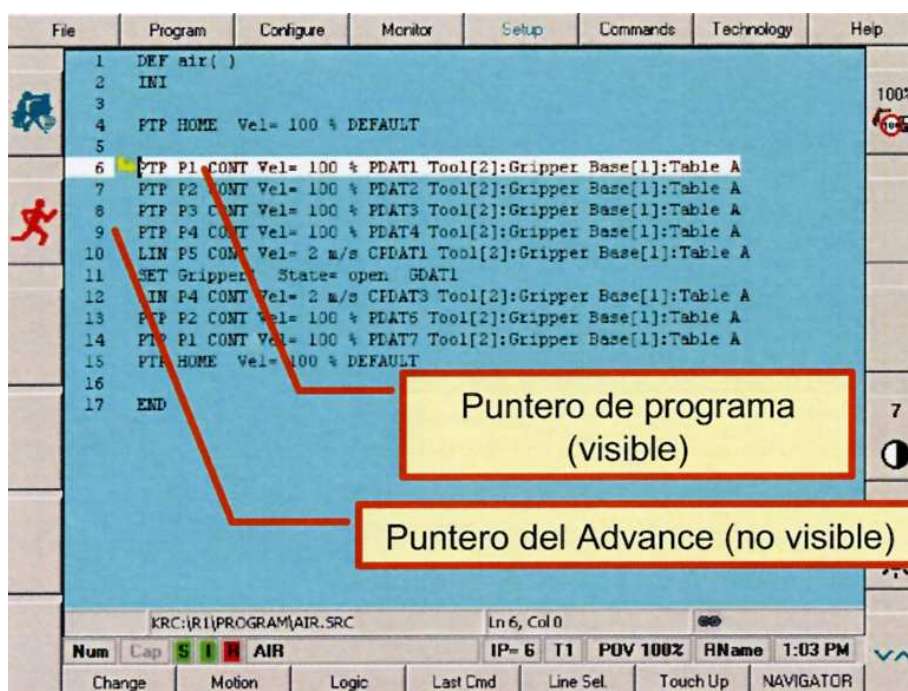


(Fig.4.5-9) Movimiento circular con contorno.

4.5.3 Concepto del Advance.

La ejecución de movimientos con aproximación va muy ligado al **Advance**, este es un puntero de programa (que representaremos con una línea blanca) en la pantalla, que indica o predice de manera futura el bloque de instrucciones que se van a ejecutar después del movimiento actual. Dicho puntero no es visible y por defecto va tres líneas de movimiento por delante del puntero de programa. Es por esto la razón por la que se pueden ejecutar los movimientos de manera continua; de una manera u otra, el sistema procesa las coordenadas de los puntos antes de que el puntero de programa se sitúe en él, y así se calcula la trayectoria futura que modifica a su vez por donde se realiza el contorno de la trayectoria actual. No solo se procesan las trayectorias antes que el puntero de programa si no que también se procesan los datos y comandos lógicos, muy a tener en cuenta a la hora de programar las condiciones lógicas, puesto que el robot podría iniciar la ejecución algún movimiento no previsto al no cumplirse la lógica cuando el puntero de programa se situase sobre la condición. Para esto se prevén instrucciones de programa que detienen el puntero del Advance.

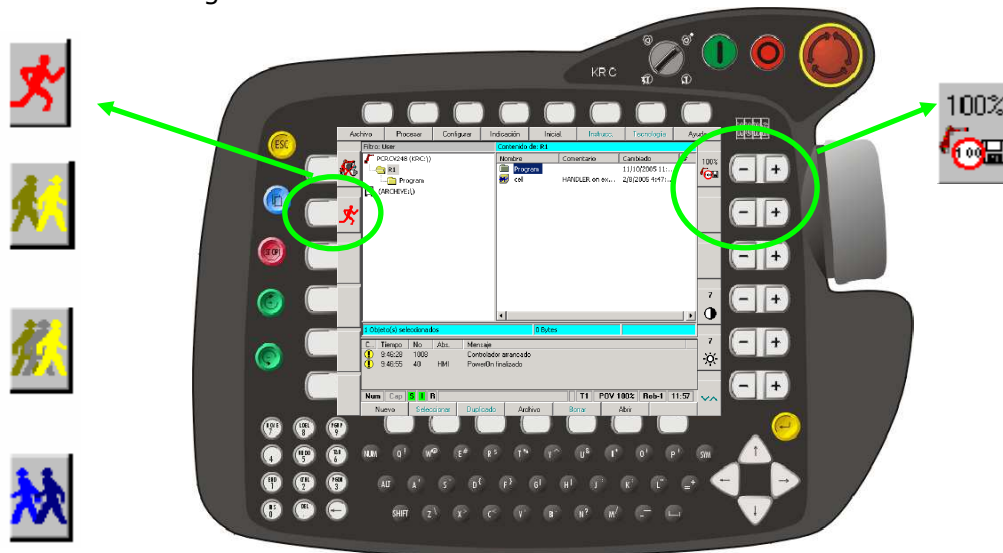
(Fig.4.5-10) Puntero del ADVANCE.



En el ejemplo mostrado cuando el puntero de programa se encuentra sobre el punto **PTP P1**, el Advance se encuentra sobre la línea **PTP P4**, de manera que el robot ya sabe la trayectoria que debe seguir tres movimientos de programa por delante; por lo tanto, y debido al cálculo de trayectorias que se procesa entre los puntos P1 a P4, la trayectoria a seguir entre P1 y P4 será lo más dinámico y suave posible. En el caso que alguno de los puntos no tuviera contorno, el Advance se detendría en dicha línea, ya que le estaríamos diciendo al robot que fuera a esas coordenadas exactas sin aproximación. En tal caso en el que el Advance es detenido se mostraría un mensaje en pantalla indicando que la aproximación no es posible.

4.5.4 Ejecutar un programa.

Antes de la ejecución deben disponerse algunos ajustes de configuración como pueden ser los siguientes:



(Fig.4.5-11) Modos de ejecución y Ajuste de velocidad en Automático

4.5.4.1 Velocidad de Programa.

Se deben efectuar pruebas de los programas reduciendo previamente la velocidad de desplazamiento del robot. Para ello se utiliza la función "**sobre control de programa**", cuya tecla se encuentra en la barra de teclas de estado de la derecha. Accionando la tecla +/-, el valor aumenta o disminuye.


4.5.4.2 Modos de ejecución de un programa.

Existen varios modos de ejecución que son los siguientes:



- **Modo Automático.** Es el modo normal de ejecución de un programa en automático, donde el puntero de programa va corriendo línea a línea de forma automática.




- **Modo Paso a Paso.** El programa es ejecutado paso a paso de punto a punto, de manera que estando con el selector en modo automático, mantendremos la tecla "**Hombre muerto**" apretando a la vez que la tecla de arranque  y volviendo a soltar para ejecutar movimiento a movimiento.

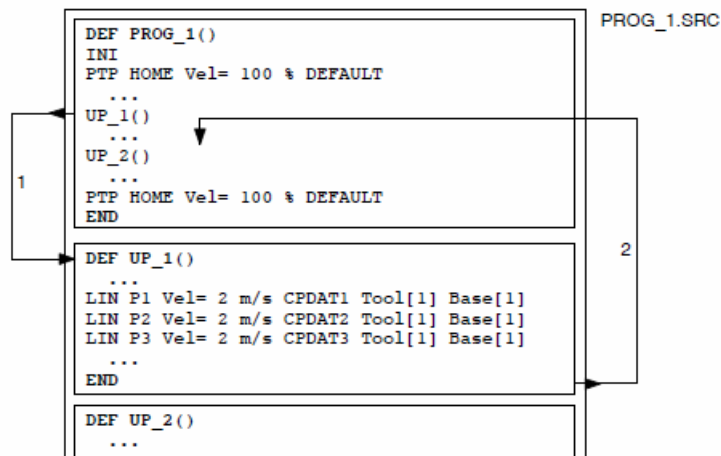


- **Modo incremental.** Disponible en modo experto, ejecuta línea a línea el programa, no saltando de un punto a otro hasta que no se encuentra en el programa una nueva definición de trayectoria.



- **Modo Paso a Paso Atrás.** Igual que el modo Paso a Paso pero en sentido inverso. 

Los modos de ejecución, nos permiten seleccionar una línea de programa e iniciar el mismo desde dicha línea, también podremos ejecutar cualquier subprograma hasta el final sin necesidad de seleccionar el principal.



(Fig.4.5-12) Ejemplo de ejecución con subprogramas

En este ejemplo en el momento que el puntero de programa pasa por el subprograma **UP_1 ()**, automáticamente se empieza a ejecutar el mismo hasta la definición **END** donde el Advance es detenido hasta que volvamos a seleccionar tecla la de arranque, donde el programa principal continuará su ejecución desde la línea en que se quedo.

Para la ejecución de líneas individuales será necesario que el selector de posición se encuentre en los modos de operación **T1 ó T2**. Para lo que se comenzará en la línea marcada con el cursor ejerciendo presión sobre la tecla de **"Hombre muerto" y "tecla de arranque"**. En cualquier momento que soltemos estas teclas la ejecución se detendrá.



4.5.4.3 La barra de estados (Interpretador SUBMIT).

La barra de estado, se encuentra presente en todo momento, y en ella se muestra la selección y estado actual del sistema de control, tal como se representa en la siguiente figura.



Podremos apreciar el estado de los modos de edición, el programa seleccionado, velocidad de programa, nombre del equipo, modo seleccionado, hora, etc.

Se nos informa mediante código de colores del estado de los accionamientos, programa e interpretador del Submit (programa cíclico multitarea) que siempre debe de estar en ejecución (verde) para que pueda funcionar todo lo demás. De este modo tenemos las siguientes indicaciones:

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	



- Interpretador del Submit no seleccionado y puesto en ejecución.



- Estado del Submit en ejecución (Programa funcional).



- Interpretador del Submit en Stop.

En interpretador del Submit, es un programa multitarea en ejecución cíclica paralelo a la aplicación que se esta ejecutando; es por lo tanto, necesario que dicho archivo se encuentre en modo ejecución para que la aplicación este funcional. En dicho interpretador se programaran aquellas tareas cuyas acciones sea necesario ejecutar constantemente, como pueden ser la interpretación de señales que nos definen en todo momento el programa de paletizado que queremos ejecutar, el número de capas, etc. Son estados de la instalación cuya lectura requiere una atención constante para el correcto funcionamiento de la misma.

Los accionamientos nos indicaran su estado activados o desactivados de la siguiente manera:



- Accionamientos en ON.



- Accionamientos en OFF.

Para el control del flujo de programa tendremos los siguientes posibles estados:



- Ningún programa se encuentra seleccionado.



- Puntero de programa se encuentra sobre la primera línea de programa.



- Programa seleccionado y en ejecución.



- Programa se encuentra detenido.



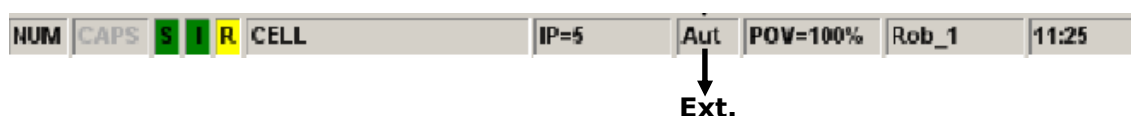
- El puntero de paso se encuentra sobre la última línea del programa seleccionado.

El puntero de paso, tal como se muestra en anteriores imágenes, se sitúa sobre la línea actual del programa que se está ejecutando. De este modo cuando queremos arrancar un programa desde el principio; y antes de ejecutar tecla de arranque, el símbolo aparecerá de color amarillo, siendo Negro cuando se ejecuta la última línea de programa antes de finalizar su ejecución. Generalmente la ejecución de programa suelen ser bucle sin fin hasta recibir la condición de salida, por lo tanto el símbolo no debiera de aparecer de color negro siempre y cuando no se produzca ninguna salida de bucle y por lo tanto finalización del programa.

4.5.4.4 Desplazamiento de coincidencia COI.

Después de seleccionar un programa, y antes de que este inicie su ejecución en cualquiera de los modos de funcionamiento, será preciso realizar un **"Punto de trayectoria alcanzada"** tal como se vio anteriormente

Una vez alcanzada la coincidencia de trayectorias, y con los accionamientos activados, ejecutaremos el programa con el selector en modo automático o automático externo, y será notificado en la línea de estados.

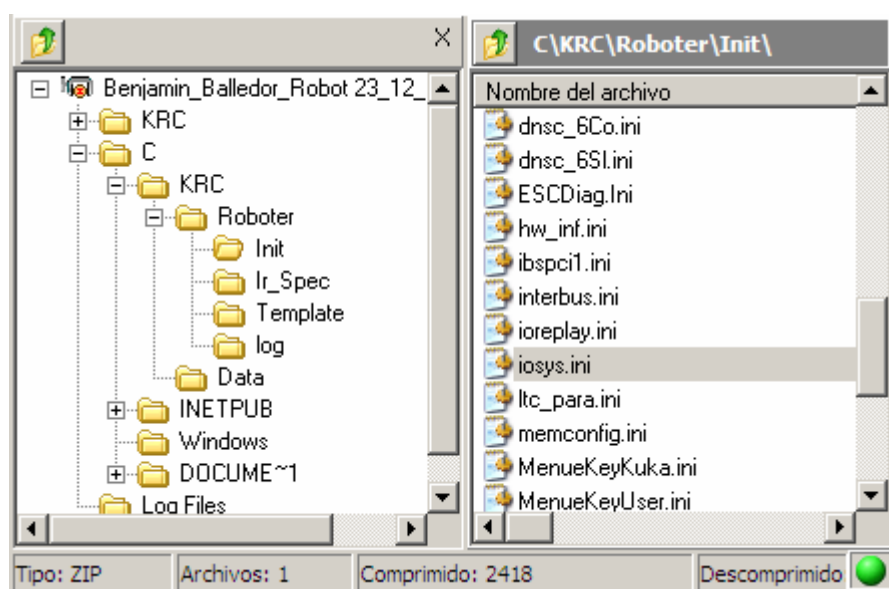


Cualquier programa que es iniciado puede ser detenido fácilmente mediante la tecla de **STOP**, en donde el puntero de programa se detendrá en ese mismo instante. Para la configuración de control por PLC o automático externo; el paró además, será controlado mediante señal.

4.5.5 Drivers E/S.

Desde la opción **Configurar->Drivers** tenemos acceso a las opciones de configuración de los protocolos de comunicación con la periferia, para transmitir vía Device Net las I/O de la periferia.

Mediante **Configurar->Drivers->Reconfiguración** abrimos el archivo **"Iosys.ini"** del sistema donde se estable el protocolo de comunicaciones y la cantidad de bytes de entrada y salida con la periferia. A este archivo también tenemos acceso directamente desde el directorio **C\KRC\Roboter\Init\iosys.ini**.



(Fig.4.5-13) Directorio del archivo iosys.ini para la activación de drivers

En este archivo se activa la comunicación por **[DEVNET]** y se determinan los bytes de entrada y salida que usaremos para la comunicación. La activación de

dicha Interface y declaración de entradas y salidas para la instalación que nos ocupa, se muestra a continuación.

```

;=====
; IOSYS.INI - Configuration file for the IO-System
;=====
; For configuration help go to the end of this file.
;-----

[DRIVERS]
;CNKE2=21,cnke2CPInit,cnke2drv.o
;DNSC6=20,dnsc6Init,dnsc6drv.o
;DNSC5=19,dnsc5Init,dnsc5drv.o
;DNSC4=18,dnsc4Init,dnsc4drv.o
;DNSC3=17,dnsc3Init,dnsc3drv.o
;CNKE1=16,cnke1CPInit,cnke1drv.o
;INTERBUSPCI=15,ibsCPPciInit,ibpcidrv.o
;DSEIO=14,dseIoInit,dseiiodrv.o
;DNSC2=13,dnsc2Init,dnsc2drv.o
;DNSC1=12,dnsc1Init,dnsc1drv.o
;PBMASL=11,pbmsInit,pfbmsdrv.o
DEVNET=2,dnInit,dn2drv.o
;INTERBUS=1,ibusInit,ibusdrv.o
;MFC=0,mfcEntry,mfcdrv.o

```

**ACTIVACION INTERFACE
DEVICE NET**

```

[DEVNET]
inb0=5,0
inb1=5,1
inb2=5,2
inb3=5,3
outb0=5,0
outb1=5,1
outb2=5,2

```

**DECLARACIÓN BYTES DE
ENTRADA Y SALIDA.**

La forma de activar dicha Interface, es quitando el ";" en la línea donde está declarado "**DEVNET=2,dnInit,dn2drv,o**" de manera que sea esta donde se declaren las señales de entradas y salidas. En el ejemplo mostrado hay configurados 4 bytes de entrada y 3 de salida a partir de la dirección 5 del módulo Beckhoff; es decir, dispondremos de 40 bits de entrada y 24 de salida.

Inb0= 5,0 ->Dirección 5 byte (entradas) 0
Inb1= 5,1 ->Dirección 5 byte (entradas) 1
Inb2= 5,2 ->Dirección 5 byte (entradas) 2
Inb3= 5,3 ->Dirección 5 byte (entradas) 3



Outb0= 5,0 ->Dirección 5 byte (salidas) 0
Outb0= 5,1 ->Dirección 5 byte (salidas) 1
Outb0= 5,2 ->Dirección 5 byte (salidas) 2

UNIDADES: 5 →

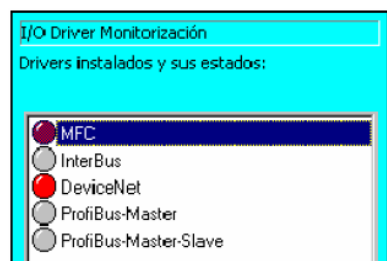
DECENAS: 0 →



(Fig.4.5-14) Dirección CAN BUS en el módulo Beckhoff

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

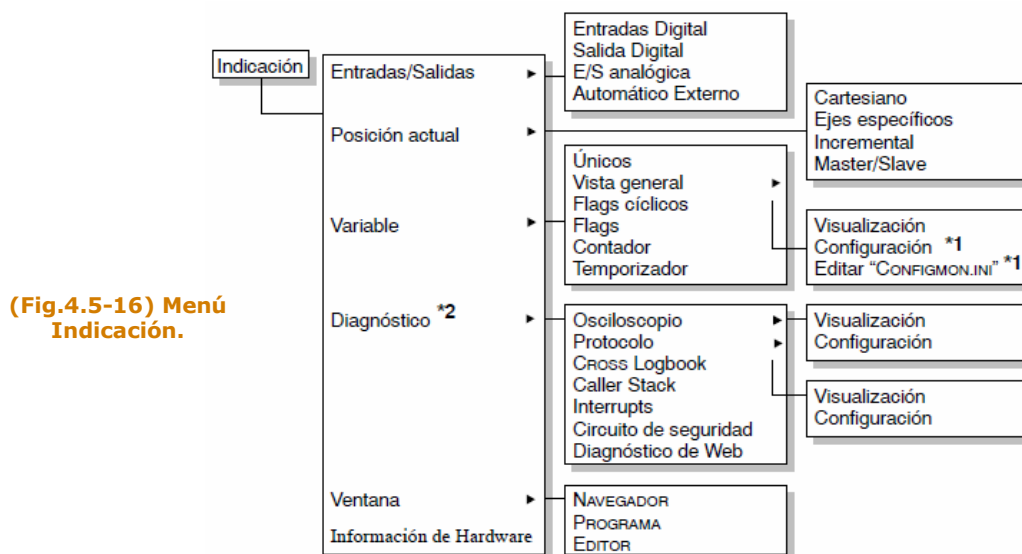
Las opciones **Configurar->Drivers->Reset** y **Configurar->Drivers->Reconfiguración E/S** del menú seleccionable con las teclas de estado, se utilizan para reconfigurar en caso de realizar cambios en este archivo, instalar y desinstalar drivers. Si monitoreamos las conexiones de esta Interface, debe mostrarse el protocolo **DeviceNet** activo a través de la tarjeta **MFC**.



(Fig.4.5-15) Reconfiguración de Drivers.

4.5.6 Menú Indicación.

Desde este menú tendremos acceso a las opciones de configuración de E/S digitales o de automático externo, podremos visualizar la posición actual de los ejes, seleccionar variables de programa y la ventana de navegación. Dentro de la ventana de programa abierta nos encontramos con la siguiente estructura:



(Fig.4.5-16) Menú Indicación.

- **Indicación->Entradas/Salidas.** Se tienen acceso a las entradas y salidas digitales configuradas, para poder ver sus estados **"true"** o **"false"**, forzado de salidas digitales y la posibilidad de modificar los textos de dichas variables. **(Remítase al apartado 4.5.1 de este mismo capítulo o al apartado 6.0 AUTOMATICO EXTERNO).**
- **Indicación->Posición Actual.** Podremos visualizar la posición de los ejes del robot si seleccionamos el modo **"Específico ejes"** o la posición con respecto a la base del robot si seleccionamos el modo **"Cartesiano"**.
- **Indicación->Variable.** Se emplea para realizar búsquedas y cambiar el estado de las variables del programa, podremos visualizar tanto las entradas

y salidas digitales, como marcas o flags cíclicos de programa (hasta 32 marcas podremos usar en el programa). Mediante la opción **Indicación->Variable->Temporizador** editaremos y configuraremos los timer que podremos usar en el programa.



(Fig.4.5-17) Configuración de temporizadores.

Disponemos de hasta 10 temporizadores editables desde la ventana y a los que seleccionaremos el valor en (ms) que activará su salida una vez habilitado desde programa. Los estados que adquieren dichos temporizadores son:

- **Rojo:** Temporizador detenido con valor ≤ 0 .
- **Rojo+Idle:** Temporizador detenido con valor >0 .
- **Verde:** Temporizador iniciado puesto a valor ≤ 0 .
- **Verde+Idle:** Temporizador iniciado con valor >0 .

4.5.7 Entradas y salidas digitales.

La manera en la que estas se visualizan en la ventana de estados es la siguiente; ambas ventanas conmutables desde las opciones de estados:

(Fig.4.5-18) Entradas y Salidas digitales

Entradas				Salidas			
	SYS	SIM	Entradas		SYS	SIM	Salidas
<input checked="" type="radio"/> 132			Entrada	<input checked="" type="radio"/> 995	SYS		Salida
<input type="radio"/> 133			Entrada	<input type="radio"/> 996	SYS		Salida
<input type="radio"/> 134		SIM	Entrada	<input type="radio"/> 997	SYS		Salida
<input checked="" type="radio"/> 135		SIM	Entrada	<input checked="" type="radio"/> 998		SIM	Salida
<input checked="" type="radio"/> 136		SIM	Entrada	<input type="radio"/> 999		SIM	Salida
<input type="radio"/> 137			Entrada	<input checked="" type="radio"/> 1000	SYS		Salida
<input type="radio"/> 138			Entrada	<input checked="" type="radio"/> 1001	SYS		Salida
<input type="radio"/> 139			Entrada	<input type="radio"/> 1002			Salida
<input type="radio"/> 140	SYS		Entrada	<input type="radio"/> 1003	SYS		Salida
<input type="radio"/> 141			Entrada	<input type="radio"/> 1004		SIM	Salida
<input checked="" type="radio"/> 142		SIM	Entrada	<input checked="" type="radio"/> 1005		SIM	Salida
<input type="radio"/> 143		SIM	Entrada				

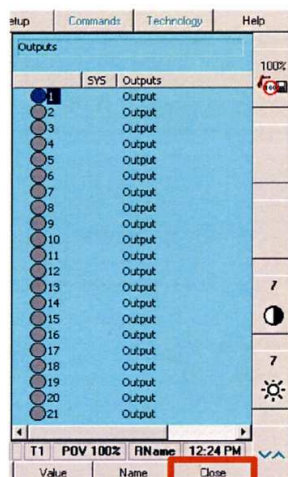
Generalmente el número máximo de entradas y salidas que podemos configurar es de 1024. El número a la izquierda, determina la entrada o salida que se está usando y la coloración si esta activada o no; es decir, true o false.



Señal activada -> true.



Señal desactivada -> false.



La denominación **SYS** indica que esa variable forma parte del sistema por lo que no podremos utilizarla de manera externa a no ser que cambiemos su configuración. La denominación **SIM** significa que dicha Entrada/Salida está siendo simulada o forzada desde la opción "**Valor**" que nos brinda la tecla de función programable.

La configuración de entradas y salidas para la comunicación del PLC mediante automático externo, se muestra de la siguiente manera:

St.	Término	Tipo	Nombre	Valor
1	Control preparado	Var	\$RC_RDY1	137
2	Alarma de paro activa	Var	\$ALARM_STOP	1013
3	Interruptor de seguridad de usuario	Var	\$USER_SAF	1011
4	Accionamientos preparados	Var	\$PERI_RDY	1012
5	Robot calibrado	Var	\$ROB_CAL	1001
6	Interficie activa	Var	\$I_O_ACTCONF	140
7	Colección de errores	Var	\$STOPMESS	1010
8	Parada de emergencia interna	Var	IntEstop	853

St.	Término	Tipo	Nombre	Valor
1	num. programa actual	Var	PGNO	0
2	Tipo de num. programa	Var	PGNO_TYPE	1
3	Ancho de bit del num. programa	Var	PGNO_LENGTH	8
4	Primer bit de num. programa	Var	PGNO_FBIT	33
5	Bit de paridad	Var	PGNO_PARITY	41
6	num. programa válido	Var	PGNO_VALID	42
7	Inicio de programa	Var	\$EXT_START	1026
8	Movimiento habilitado	Var	\$MOVE_ENABLE	1025
9	Confirmación de error	Var	\$CONF_MESS	1026
10	Accionamientos desconectados (in)	Var	\$DRIVES_OFF	1025
11	Accionamientos conectados	Var	\$DRIVES_ON	140
12	Activar interficie	Var	\$I_O_ACT	1025

(Fig.4.5-19) Configuración de entradas y salidas para la Interface con automático externo.

En la columna **St.** Se muestra el estado actual de las señales, siendo true o false como en el caso anterior. La diferencia principal radica en que podremos encontrar dos tipos de señales siendo estas variables o I/O digitales.

Var **Variable.** Son marcas de sistema para el control, cuyo valor podemos leer ni escribir.

Var **I/O.** Entradas y salidas físicas para la comunicación del PLC con el KRC. Dentro de este apartado encontramos las entradas y salidas digitales que usaremos para manejar el robot desde el PLC, como pueden ser los modos de operación en la que se encuentra el selector, determinados por las variables **\$T1**, **\$T2**, **\$AUT (Automático)** o **\$EXT (Externo)**, así como la confirmación de los accionamientos que sería **\$DRIVES_ON** o **\$DRIVES_OFF**, etc.

Valor hace referencia al número de entrada/salida o canal, que al igual que en el caso anterior direccionaremos para que puedan ser utilizadas por el PLC. En cualquier caso encontraremos más detalles sobre la configuración del automático externo en el apartado correspondiente.

4.6 PROGRAMACIÓN LÓGICA.

Se explicará en este apartado las sintaxis y funciones básicas de programación para la edición de un programa.

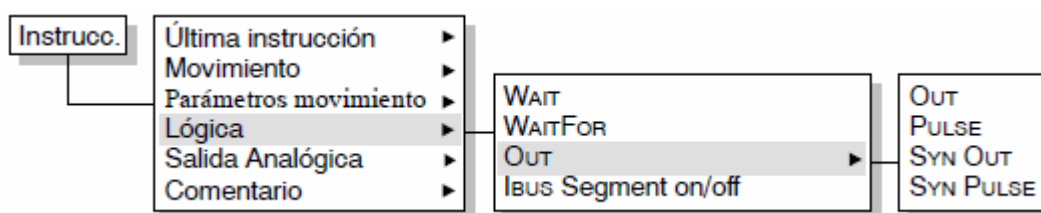
4.6.1 Grupos de Usuarios.

Existen tres grupos definidos de usuarios:

1. **Usuario.** Sin derechos de programación, son usados para el cliente final, manejo y selección de programas. Este será el grupo de usuario desde donde arrancará el sistema por defecto.
2. **Experto.** Puede modificar y programar con la sintaxis correspondiente, así como editar los archivos de configuración.
3. **Administrador.** Para cambiar el grupo de usuario lo haremos desde el menú **Configurar-> Grupo de Usuario**. Seguidamente se nos desplegará un formulario donde tendremos que introducir la contraseña para validar el cambio. Por defecto en los sistemas nuevos; y si no se ha hecho ningún cambio, la contraseña que viene preestablecida será **"Kuka"** en minúsculas.

4.6.2 Menu Instrucción.

Con la ventana de programa abierta, tenemos acceso a la siguiente lista de instrucciones que nos facilitará la programación lógica y nos proporcionarán herramientas de trazado de trayectorias.



(Fig.4.6-1) Menú Instrucción

- **Instrucción->Última instrucción.** Mediante este comando podremos repetir la última instrucción de programa ejecutada, ya sea un punto o una función de programa.
- **Instrucción->Movimiento.** Para la declaración de movimientos tipo **PTP**, **LIN** o **CIRC**.
- **Instrucción->Parámetros de Movimiento.** Permite la modificación del entorno tubular de control para el control contra colisiones.
- **Instrucción->Lógica.** Permite la entrada de instrucciones de espera, y condiciones de salidas en función de la trayectoria.

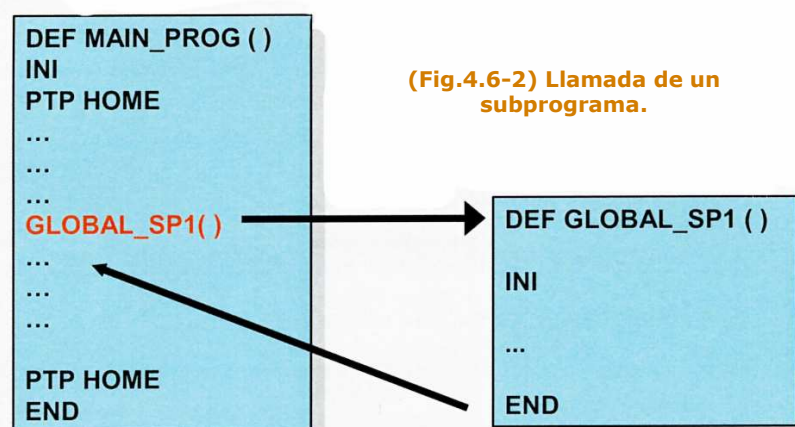
4.6.3 Estructura de un programa.

Un programa se compone básicamente de un encabezado con el nombre del programa, seguidamente las declaraciones, el cuerpo del programa (instrucciones) y la finalización.

```

DEF NAME (máximo 24 caracteres)
Declaraciones
;===Comentarios===
.....
Inicialización de variables
;===Comentarios===
Instrucciones
;===Comentarios===
END

```



La llamada a un subprograma se realiza únicamente poniendo el nombre del programa seguido de los paréntesis. Una vez que esta ha sido procesada, se vuelve a la línea siguiente de programa donde se realizó la llamada, la cantidad máxima de llamada a subprograma dentro de subprograma es de hasta 20.



Las declaraciones se evalúan antes de la ejecución del programa y no tienen el carácter dinámico que tienen las instrucciones en constante ejecución.

Otra parte importante en la estructura de programa son los comentarios, que no influyen en el ciclo de ejecución del programa y ayuda a esclarecer y comprender el mismo. Estos siempre empiezan con el punto y coma ";".

Existen también lo que es conocido como "**Llamadas a función**" en las que se insertan unos parámetros y esta nos devuelve un valor, no se detallará su funcionamiento en este manual por no carecer de importancia dentro del programa de paletizado que nos ocupa.

4.6.4 Declaración de Variables.

La declaración de variables se representa con un número máximo de 24 caracteres, y no deben empezar por un valor numérico. Todas las variables del

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

sistema comienzan mediante el carácter "\$". Como ya comentamos al inicio de este documento, se trata de desarrollar documentación explícita con respecto a la instalación que nos ocupa; en este sentido, no se detallarán aquellas variables que no tengan uso por no manejarse en el presente proyecto, como pueden ser manejo de matrices, estructuras, etc.

Existen cuatro tipos de variables simples que contienen un único valor, y todas ellas se declaran anteponiendo la denominación **"DECL"**.

DECL + "Tipo de Variable" + "Nombre"

Una vez definida la variable, su posición en memoria se define como no válido, por lo que es necesario asignarles un valor inicial. Tómese por ejemplo la declaración de variables del archivo **"\$CONFIG.DAT"** a las que se les asigna un valor por defecto mediante el símbolo de igualdad **"="**.

4.6.4.1 Variable entera "INT".

Como dice su nombre, sólo admite datos enteros, produciéndose un redondeo en el caso de asignaciones con números reales. Admite hasta 32bit incluido el bit de signo, por lo que se le puede asignar el rango de valores +/- 2e31: es decir +/- 2147483647.

DECL INT Numero = 0

Declaramos la variable entera **Numero** y le asignamos el valor **"0"**. En este momento dicha variable podrá cambiar su valor en el programa cuantas veces sea necesario.

4.6.4.2 Variable real/booleana/carácter "REAL/BOOL/CHAR"



La declaración de este tipo de variables se produce de manera semejante a la variable entera, diferenciando que los valores reales admiten coma flotante, el valor booleano se utiliza para diferenciar estados lógicos; pues sólo admiten dos tipos de valores **"true"** o **"false"** y la variable carácter representa un valor del código o tabla **ASCII**.

DECL REAL Numero = 0,5
DECL BOOL Estado = false
DECL CHAR Letra = "a"

4.6.4.3 Variable de tipo Estructura "STRUC".

Su uso en este manual, es simplemente la definición de variables que contienen varios tipos de datos. En este caso se emplea por ejemplo para determinar la posición de los ejes, este tipo de datos consta de 6 valores reales y 2 de tipo entero que son definidos internamente en los archivos de sistema:

STRUC POS REAL X, Y, Z, A, B, C, INT S, T

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

4.6.4.4 Variables "SIGNAL".

Las variables de tipo "**SIGNAL**" toman el valor decimal de la combinación binaria de las entradas lógicas que las componen, o la declaración de variables enteras **\$INT** , que tomando un valor numérico en concreto.

Así por ejemplo en **\$CONFIG.DAT** se definen las variables de tipo **SIGNAL registro_1 y registro_4** compuestas por la combinación binaria de la entradas lógicas **\$IN [21] ..\$IN[24]**.

SIGNAL registro_1 \$in[21] TO \$in[24]
SIGNAL registro_4 \$in[21] TO \$in[24]

Registro_1 tomará el valor de 0 a 16, que son las combinaciones binarias que se pueden hacer con las cuatro entradas **\$IN[21], \$IN[22], \$IN[23] y \$IN[24]**.

4.6.4.5 Variables de sistema.

Permiten la programación cómoda de los sistemas robotizados en el lenguaje del robot, estas pueden ser leídas y escritas como cualquier variable de programa. Algunos ejemplos de variables predefinidas que destacamos son los siguientes:

4.6.5 OPERADORES.

Para la manipulación de datos y las operaciones, se hace uso de los denominados operadores aritméticos. La mayoría de estos operadores y símbolos son los habituales en cualquier lenguaje de programación.

4.6.5.1 Operadores aritméticos.

- **"+"**. Suma o signo positivo.
- **"-"**. Resta o signo negativo.
- **"*"**. Multiplicación.
- **"/"**. División.



4.6.5.2 Operador Geométrico ":"

- **":"**. Operador geométrico.

El operador geométrico se utiliza para realizar asignaciones entre variables de tipo **"FRAME"** o tipo **"POS"**. Así por ejemplo a una variable de tipo **FRAME** le puedo asignar las posiciones de una de tipo **POS** dando como resultado otra variable de tipo **POS** sin que las variables **S,T** se vean afectadas.

FRAME BASE, SUMA1, SUMA2 Defino tres variables de tipo **FRAME**.
SUMA1 = {X 450,Y 600,Z 800,A 0,B 0,C 0}
SUMA2 = {X 80,Y 110,Z 55,A -40,B 180,C 0}

BASE = SUMA1:SUMA2
BASE = {X 530,Y 710,Z 855,A -40,B 180,C 0}

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

4.6.5.3 Operadores de comparación.

Se utilizan para formar expresiones lógicas cuyo resultado sólo puede tener dos valores, cierto **"true"** o falso **"false"**.

- **"=="**. Operador de igualdad.
- **"<>"**. Operador de desigualdad.
- **">"**. Mayor que.
- **"<"**. Menor que.
- **">="**. Mayor o igual que.
- **"<="**. Menor o igual que.

4.6.5.4 Operadores lógicos.

Permiten la combinación de variables booleanas y expresiones lógicas formadas mediante los operadores de comparación. Estos son los siguientes:

- **"NOT"**. Negación o inversión lógica.
- **"AND"**. Y Lógico.
- **"OR"**. O Lógico.
- **"EXOR"**. O Exclusivo

Estos mismos operadores pueden hacerlos operando sobre **bit** de datos en vez de con expresiones lógicas, para ello bastaría anteponerles el carácter **"B"** sobre el nombre de la variable dando como resultado: **B_NOT,B_AND,B_OR,B_EXOR**.

4.6.6 Instrucciones de Programa.

Se definirán en este apartado la sintaxis para el control de programa que es necesaria aplicar para la instalación de robot paletizado, para ello destacaremos las siguientes instrucciones de programa.

4.6.6.1 Rama Condicional **"IF / ENDIF"**.

La sintaxis es la siguiente:

IF Condición THEN
Sentencia
ELSE
Sentencia
ENDIF

Dependiendo de la condición se ejecutará el bloque de sentencias THEN , o la alternativa ELSE. Cada instrucción IF debe cerrarse con la sentencia ENDIF.

4.6.6.2 Distribuidor "SWITCH / ENDSWITCH"

```

SWITCH Variable

    CASE 1
        Sentencias
    CASE 2
        Sentencias
    ...

    DEFAULT

ENDSWITCH

```

La instrucción de distribución es una instrucción de selección para varias ramas del programa, en la que solamente se ejecuta una rama para saltar inmediatamente a la declaración ENDSWITCH.

El número de CASE existentes dependerá de la variable del SWITCH, en cualquier caso de no cumplirse el CASE se ejecutará DEFAULT.

4.6.6.3 Bucles sin fin "LOOP / ENDLOOP".

```

LOOP

    Sentencia 1

    ...

    Sentencia n

ENDLOOP

```

Los bucles son ejecuciones cíclicas de programa, en donde todo el código de programa que hay en el interior se repite continuamente. En caso de querer terminar con dicho bucle se ha de llamar a la sentencia **EXIT**.

4.6.6.4 Instrucción "HALT".

```

DEF zeros( )
INI

    ptp {a1 -45,a2 -90,a3 90,a4 0,a5 +90,a6 0}
    halt
    PTP HOME Vel= 100 % DEFAULT

    ptp {a1 0,a2 -90,a3 90,a4 0,a5 0,a6 0}
    ;posicion mordaza
    ptp {a1 0,a2 -90,a3 90,a4 0,a5 +90,a6 0}
    halt



    PTP HOME Vel= 100 % DEFAULT

END

```

Esta llamada se utiliza para detener la ejecución de un programa, y continuará si es nuevamente accionada la tecla de arranque. Muy útil si queremos hacer un programa que lleve el robot a una posición determinada como puede ser el zero mecánico. Tómese como ejemplo el siguiente programa:

En el programa con nombre **zeros ()** se realizan varios movimientos donde se lleva el robot a unas determinadas posiciones.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

5 AUTOMATICO EXTERNO (PLC).

En términos generales, lo que se va a tratar en este apartado son las configuraciones necesarias que se precisan hacer para que el sistema KRC pueda ser controlado mediante PLC, y el intercambio de señales que necesitan ser declaradas y asociadas al PLC para tal fin.

"CELL.SRC" es el programa encargado de la organización de las aplicaciones; de paletizado en este caso, que se han de ejecutar, y todo ello mediante asignación de entradas y salidas físicas a variables de sistema para el control del robot.

Lo primero que debemos hacer en este caso es configurar la Interface de señales desde el menú **[Configurar > Entradas/Salidas > Automático Externo]** y en la que nos encontraremos las opciones de configuración de entradas y salidas:

Una de las señales que tendremos que declarar para activar la Interface es la variable de sistema **\$IO_ACT**, de manera que si a través de PLC la activamos estaremos estableciendo comunicación con el KRC, y así podremos iniciar la ejecución del programa **CELL.SRC**.

El programa de comunicaciones, cuenta en la cabecera con la secuencia de inicialización, donde están declarados los programas que desde cell.src se van a ejecutar de manera continua. Estos son los siguientes:

```
DEF CELL ( )
EXT mosaico_5 ( )
EXT mosaico_4 ( )
EXT mosaico_3 ( )
EXT home ( )
```

En este sentido cobra vital importancia las líneas de comando donde se verifica la posición segura del robot.

```
home ( )
;FOLD CHECK HOME
$H_POS=XHOME
IF CHECK_HOME==TRUE THEN
P00 (#CHK_HOME,#PGNO_GET,DMY[,0] );Testing Home-Position
ENDIF
```

"HOME ()" ; tal como detallaremos más adelante, es el subprograma que se ejecutará cada vez que entremos en CELL.SRC en donde se asegura un punto seguro mediante COI alcanzada y se determina el número de sacos que se van a paletizar en función del programa y el número de capas que seleccionemos. Una vez que el punto de trayectoria seguro está alcanzado y se ha seleccionado un programa MOSAICO_3 (), MOSAICO_4 () o MOSAICO_5 () se activará el módulo de ejecución POO.SRC. Hay que decir que este punto de programa HOME (), sólo se ejecutará cada vez que se seleccione el programa CELL.SRC y se inicie desde la consola KCP, esto ocurrirá por ejemplo cada vez que se reinicie el equipo haciendo un ARRANQUE EN FRÍO, donde será el operario quien tendrá que seleccionar dicho módulo para que comience su ejecución. Una vez que CELL.SRC está en ejecución y mientras no se reinicie el equipo desde cero, este no tendrá que volver a ser ejecutado desde la KCP, estando el control supeditado al PLC.

El módulo de activación **"POO.SRC"** al que no tendremos que realizar ninguna modificación, es el que nos permitirá (de manera oculta a nuestros ojos) tener acceso al CELL.SRC dentro de su bucle de ejecución, para cambiar el programa de paletizado que se desea usar dependiendo de la función **"Switch"**, donde se determina a través de las variables de sistema **"PGNO_BIT"** (que veremos en la declaración de entradas) el programa **"CASE"** que se va a ejecutar. P00 es por lo tanto, donde se encuentran las funciones necesarias para transferir los números de programa a través de un ordenador principal o PLC.

LOOP

```

P00 (#EXT_PGNO,#PGNO_GET,DMY[,0 )
SWITCH PGNO ; Select with Programnumber
CASE 1
  P00 (#EXT_PGNO,#PGNO_ACKN,DMY[,0 ) ; Reset Progr.No.-Request
  mosaico_5 ( ) ; Call User-Program
CASE 2
  P00 (#EXT_PGNO,#PGNO_ACKN,DMY[,0 ) ; Reset Progr.No.-Request
  mosaico_4 ( ) ; Call User-Program
CASE 3
  P00 (#EXT_PGNO,#PGNO_ACKN,DMY[,0 ) ; Reset Progr.No.-Request
  mosaico_3 ( ) ; Call User-Program
;DEFAULT
  ;P00 (#EXT_PGNO,#PGNO_FAULT,DMY[,0 )
ENDSWITCH
ENDLOOP

```

El programa de ejecución CELL.SRC estaría en el bucle infinito **"LOOP"** de no haber ningún programa seleccionado, cosa que hemos hecho en **"HOME ()"**.

5.1 SEÑALES DE ENTRADAS.



TitleConfigInput				
	ColTitleName	title	Nombre	Valor
1	PGNO_TYPE	Var	PGNO_TYPE	1
2	REFLECT_PROG_NR	Var	REFLECT_PROG_N	0
3	PGNO_LENGTH	Var	PGNO_LENGTH	8
4	PGNO_FBIT	NO	PGNO_FBIT	33
5	PGNO_PARITY	NO	PGNO_PARITY	41
6	PGNO_VALID	NO	PGNO_VALID	42
7	\$EXT_START	NO	\$EXT_START	1026
8	\$MOVE_ENABLE	NO	\$MOVE_ENABLE	1025
9	\$CONF_MESS	NO	\$CONF_MESS	1026
10	\$DRIVES_OFF	NO	\$DRIVES_OFF	1025
11	\$DRIVES_ON	NO	\$DRIVES_ON	140
12	\$I_O_ACT	NO	\$I_O_ACT	1025

(Fig.5.1-1) Señales de entrada para el automático externo

Nos encontramos una ventana con las siguientes características en las que se hace una diferenciación; como bien comentamos anteriormente, sobre las variables del sistema y las I/O ; en este caso entradas ,que se manejan (Fig.9.1-1).

Desde este mismo menú tendremos acceso mediante teclas de función, para cambiar la visualización a la configuración de las salidas, así como para cambiar el direccionamiento de las variables mediante la tecla **"Valor"**. Se usarán las siguientes señales:

- **PGNO_TYPE:** Se utiliza como variable de sistema y en ella configuraremos la forma en la que se leerá el número de programa siendo este un valor binario **PGNO_TYPE = 1**.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

1. BINARIO: PGNO_TYPE = 1 será la forma en la que transmitiremos desde el PLC el nº de programa.
2. BCD: PGNO_TYPE = 2

- **PGNO LENGHT:** En esta otra variable indicaremos el número de bits que emplearemos para transmitir el programa **PGNO LENGHT=3**. En nuestro caso tenemos 3 programas que serían los siguientes:

1. MOSAICO_5. Será el número de programa 1.
2. MOSAICO_4. Será el número de programa 2.
3. MOSAICO_3. Será el número de programa 3.

Para escribir en binario hasta 3 solamente necesitamos dos bits, aunque en nuestra instalación con la previsión de añadir más programas en el futuro se disponen de 3 bit para seleccionar programas.

- **PGNO_FBIT:** Se le indicará el primer bit a partir del cual se le pasará el programa. El programa será transferido en binario desde el PLC a través de las entradas **\$IN[1], \$IN[2] y \$IN[3]**, por lo tanto **PGNO_FBIT = 1**.
- **PGNO_PARITY:** Indicamos el bit de paridad o si este no se evalúa. En nuestro caso no usaremos bit de paridad por lo tanto **PGNO_PARITY = 0**

1. PGNO_PARITY. Un valor positivo significa paridad PAR.
2. PGNO_PARITY. Un valor negativo significa paridad IMPAR.
3. PGNO_PARITY. Valor 0 significa sin evaluar.

- **PGNO_VALID:** Será la entrada que habrá que habilitar desde el PLC, para indicar a la unidad de control el comando de lectura de programa. **PGNO_VALID = 5**.
- **EXT_START:** Se evalúa el flanco creciente de la señal, y se utiliza para avisar a la unidad de control que queremos realizar un cambio de programa, o activar CELL.SRC. Esta señal de entrada se encuentra en la entrada del robot **\$IN[6]**, por lo tanto **EXT_START = 6**.



- **MOVE_ENABLE:** En automático externo esta señal funciona al igual que el arranque y parada de programa **"START", "STOP"**. Si la señal se encuentra en **"true"** significa que se pueden realizar movimientos y continuar con la ejecución del programa, **"false"** se detienen los accionamientos de manera controlada. Esta señal se encuentra direccionada en la **\$IN[7]**, por lo tanto **MOVE_ENABLE = 7**.

- **CONF_MESS:** Se utilizará para confirmar los mensajes de la pantalla de mensajes de la KCP. **CONF_MESS = 8**.



- **DRIVES_ON:** Misma función que los accionamientos en ON de la KCP. Se precisará un impulso de al menos 20ms para activar los accionamientos. **DRIVES_ON = 9**.



- **DRIVES_OFF:** Mediante un impulso de nivel bajo de unos 20ms, se pueden desactivar los accionamientos. Está entrada tendrá por lo tanto, que permanecer a nivel alto. **DRIVES_OFF = 10.**

5.2 SEÑALES DE SALIDA.

TitleConfigOutput				
	ColTitleName	itleT	Nombre	Valor
1	\$RC_RDY1		\$RC_RDY1	137
2	\$ALARM_STOP		\$ALARM_STOP	1013
3	\$USER_SAF		\$USER_SAF	1011
4	\$PERI_RDY		\$PERI_RDY	1012
5	\$ROB_CAL		\$ROB_CAL	1001
6	\$I_O_ACTCONF		\$I_O_ACTCONF	140
7	\$STOPMESS		\$STOPMESS	1010
8	PGNO_FBIT_REFL		PGNO_FBIT_REFL	999

Para las salidas, la pantalla que se nos muestra es similar, pero nos aparecen pestañas de selección con las que podremos ir configurando las diferentes variables y salidas de sistema, cosa que podremos hacer mediante las teclas de tabulación "Tab +" y "Tab -"

(Fig.5.1-2) Señales de salida para el automático externo.

Las señales a manejar serán las siguientes:

- **STOMESS:** Señal de salida utilizada por la unidad de control KRC, para indicar la existencia de mensajes pendientes por confirmar que han podido provocar anteriormente el paro robot. **STOPMESS = 3.**
- **PGNO_REQ:** Se le indica al PLC mediante un cambio de señal que ya se puede transmitir un programa. **PGNO_REQ = 5.**
- **APPL_RUN:** Con esta señal activada, el KRC indica al ordenador principal que se está ejecutando un programa. **APPL_RUN = 6.**
- **PERI_RDY:** Es la señal con la cual el robot indica al ordenador principal que los accionamientos están activados. **PERI_RDY = 1.**
- **ALARM_STOP:** Indica que se ha producido una parada de emergencia. **ALARM_STOP = 7.**
- **T1,T2,AUT,EXTERN:** Son señales de salida que nos indica el estado del selector de movimientos. Por utilidad dos señales para activar el estado "T1" **Movt.Manual** y otra que por seguridad nos indique que el robot se encuentra en modo externo "EXTERN".

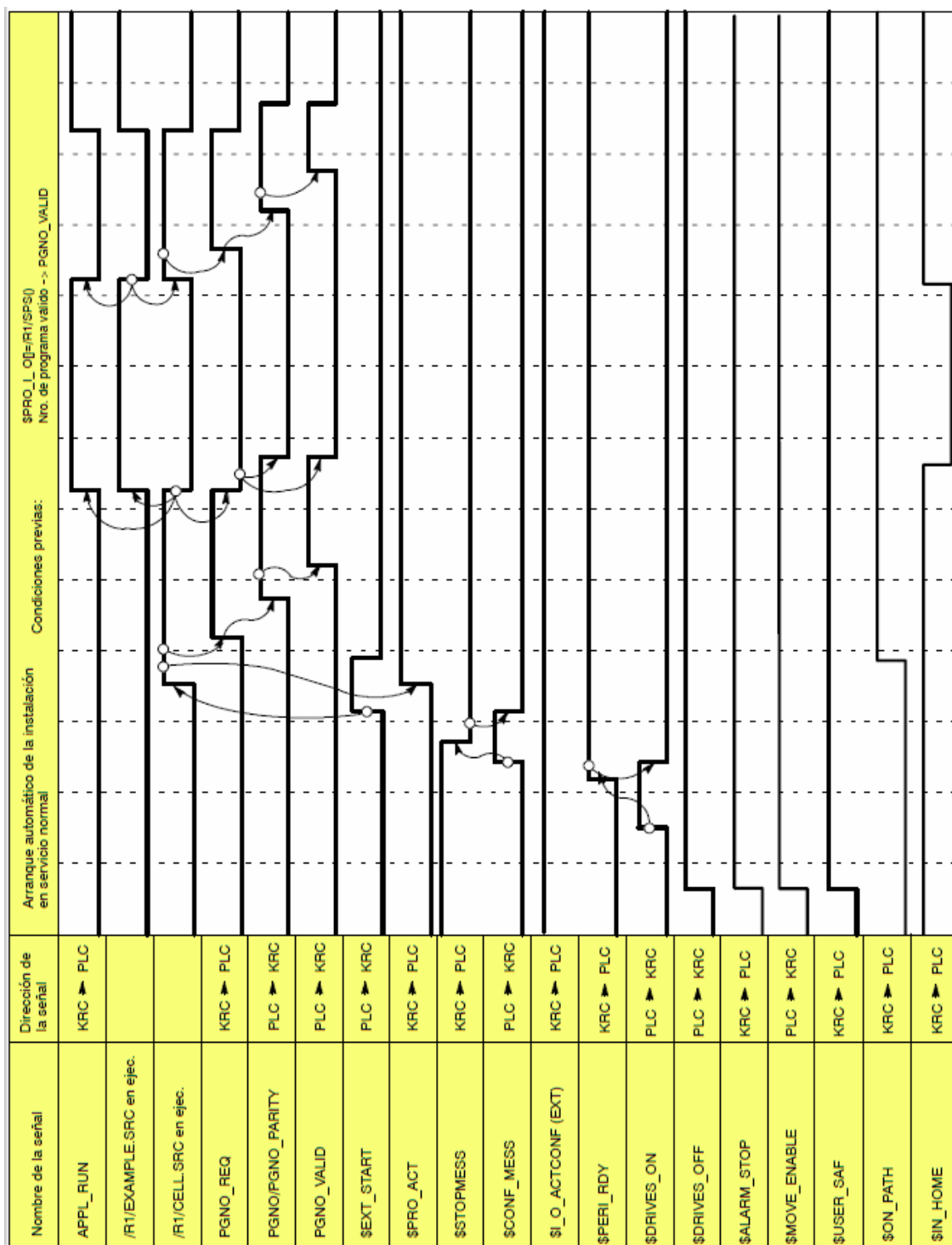
1. **Modo T1 = 8 -> \$OUT[8]**



2. **Modo EXTERN = 9 -> \$OUT[9]**

- **PRO_ACT:** Esta señal; a diferencia de APPL_RUN, estará activa siempre y cuando haya activo algún proceso activo en el sistema. **PRO_ACT = 4.**

5.3 DIAGRAMA DE SEÑALES.

Por su significado relevante, se adjuntará el diagrama de señales donde se detalla todo el proceso de Interface necesario para manejar el robot.



	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

En el diagrama adjunto se muestra la maniobra y condiciones que se deben cumplir en cuanto a las señales de Interface. Para ello deben cumplirse lo siguiente:

1. Move Enable. Debe ser mantenida por el PLC a nivel alto '1'. Para ello será condición indispensable que el selector de modos se encuentre en modo Automático externo, **EXTERN = '1'** o manual **T1 = '1'**.

2. Una vez que tenemos el movimiento habilitado y siempre y cuando no haya ninguna seguridad de robot activada **ALARM_STOP = '1'** podremos activar los accionamientos. **DRIVES_OFF = '1'** permanecerá a nivel alto, y **DRIVES_ON** será programado mediante temporizador para provocar un pulso de 200ms para asegurar la lectura por el KRC.

3. En el flanco de subida del pulso anterior, el PLC esperará la confirmación de la señal **PERI_RDY = '1'** que indicará manteniendo esta señal que dichos accionamientos están activados.

4. Con Los accionamientos activados, en caso de que el KRC indique la existencia de mensajes por confirmar **STOPMESS = '1'**, el PLC activará la señal **CONF_MESS = '1'** y esperará a que STOPMESS cambie de nivel **STOPMESS = '0'**.

5. Una vez confirmados los mensajes el PLC desactivará CONF_MESS y activará mediante pulso de 20ms la señal **EXT_START = '1'** para que de comienzo la ejecución del programa **CELL.SRC**.



6. Cuando Cell arranca, el KRC activará y mantendrá la señal **PRO_ACT = '1'** para indicar que un proceso se está ejecutando, de esta manera el PLC desactivará la señal de EXT_START. Seguidamente la unidad de control; y a través de la señal **PGNO_REQ = '1'**, indica al PLC que puede transmitir el número de programa.

7. El equipo del PLC levantará la señal **PGNO/PGNO_PARITY** para la preasignación del programa transcurrido un tiempo desde el consentimiento a través de PGNO_REQ, para seguidamente levantar **PGNO_VALID = '1'** dando permiso al KRC para realizar la lectura del programa que se debe ejecutar.

8. La unidad de control leerá el número de programa transmitidos a través de los tres bits de programa por el PLC, una vez realizada la lectura el KRC volverá a resetear la señal **PGNO_REQ = '0'** y se activará la señal **APPL_RUN = '1'** con la ejecución del programa ejemplo (en este caso de paletizado). Transcurrido un tiempo PGNO y PGNO_VALID vuelven a su estado de esposo.

Observar que cada vez que finalicemos la ejecución del programa de paletizado, de manera automática se volverá a ejecutar el programa CELL.SRC volviendo a realizar la petición de nº de programa a través de la señal PGNO_REQ.

Los accionamientos y confirmación de mensajes no serán necesarios realizarlos mientras no se produzca ninguna para externa. En cualquier momento que se produzca un paro de la marcha del robot provocado por la periferia y PLC, se reiniciarán todas las señales para asegurarnos que en caso de precisar una nueva lectura de programa se haga el ciclo completo de señales.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

6 SOFTWARE DE LA INSTALACIÓN.

Se describirá en este capítulo, la sintaxis y lógica de funcionamiento de los programas del **Robot** y **Autómata** de forma detallada, abarcando cada uno de los puntos de funcionamiento y acciones llevadas a cabo. Para ello nos ayudaremos de los característicos diagramas de flujo que representarán de manera gráfica las secuencias y pasos necesarios para realizar la paletización.

A nivel software comenzaremos realizando una descripción detallada abarcando el tema de configuración de mosaicos, ya que las definiciones descritas serán muy útiles; pues se hace constante referencia a ellas. Una vez terminada esta descripción se tratarán los procesos existentes bien diferenciados a llevar a cabo en la instalación y en los cuales subdividiremos este capítulo:



1. Configuración de Mosaicos.
2. Programa de paletizado mediante lenguaje y sintaxis de programación propia de los sistemas KUKA KRC-2.
3. Desarrollo y manejo de la periferia estableciendo comunicación y control con el sistema KRC-2.

Para el apartado número 2 se presentará de manera general el proceso cíclico de ejecución principal, estableciendo los puntos de conexión o comunicación con el autómata. Después se entrará con mayor detalle en cada principio lógico que forman el conjunto de programa. Para el segundo apartado se detallará mediante diagramas los procesos llevados a cabo mediante lenguaje de contactos. Este tipo de software se encuentra subdividido en secciones que identifican claramente cada uno de los procesos llevados a cabo por el PLC.

6.1 CONFIGURACIÓN DE MOSAICOS.

Mostraremos a modo de representación, como se realiza el dibujo a paletizar en los diferentes tipos de palet's empleados en "**Piensos O' Couto**". Anteriormente se habló de los tipos de palet's que se emplearán en la instalación y el dibujo o mosaico que se paletizará, que dependerá del número de sacos por capa.

Son tres los programas de paletizado que se realizará, todos ellos son configurables en cuanto al número de capas o alturas que queremos realizar, sin embargo no es configurable en cuanto al peso del saco, ya que los puntos de dejada al paletizar un mosaico de 5 sacos por capa no es el mismo si el saco pesa 15Kg. que si pesa 40Kg., donde los puntos de dejada debieran abrirse más para que no cayera uno encima de otro. Esto implicaría realizar un programa de paletizado por cada mosaico y por cada pesaje diferente, donde podríamos llegar a tener hasta 12 programas de paletizado diferentes para los 12 tipos de sacos y pesajes diferentes que pudiera tener el cliente. En este aspecto lo acordado con "**Piensos O' Couto**" fue la definición únicamente de tres tipos de programa de paletizado, cada uno realizado para un tipo de pesaje en concreto. De esta manera los programas a realizar serían:

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

- 1. Mosaico de tres sacos.** Para sacos de 40Kg. bajo palet de 800*1000.
- 2. Mosaico de cuatro sacos.** Para sacos de (30 ó 40) Kg. bajo palet de 1000*1200.
- 3. Mosaico de cinco sacos.** Para sacos de (25 ó 30) Kg. bajo palet de 1000*1200.

Todas las demás configuraciones que salgan de estos estándares correrán bajo la responsabilidad de **"Pensos O' Couto"**, ya que si el cliente quiere paletizar sacos de 40Kg. con el mosaico de 5 sacos, no se asegura bajo ningún concepto que el palet quede simétrico o no se produzcan caídas de los mismos.

Antes de abordar los mismos, conviene aclarar algunas definiciones y conceptos necesarios.

- **ALTURA.** Se define **"Altura"** como el número de filas o alturas máximas que alcanzará un palet. Para **"Pensos O' Couto"** el número máxima de alturas que podemos configurar es de 10, esto es debido a limitación físicas de los ejes del robot, en cuanto a la altura máxima que es capaz de apilar y las medidas máximas del espesor del saco (Fig.6.1-1).



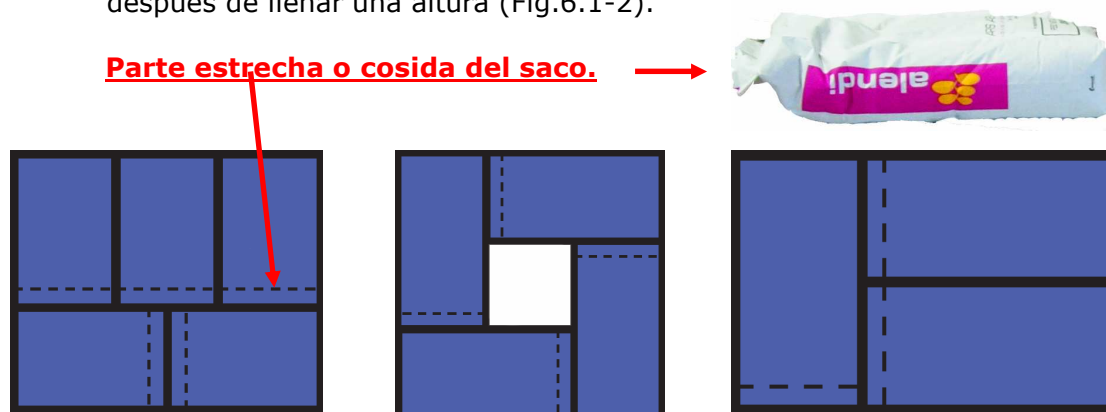
(Fig.6.1-1) Altura máxima de apilamiento

El Robot en cada altura se situará 190mm (el espesor del saco) por encima. A este valor se le añade un $\pm \Delta_{\text{Altura}}$ (Incremento de altura) configurable desde la **XBT N401**, donde se le da al operario la posibilidad de aumentar o disminuir ligeramente este valor $\pm 350\text{mm}$. Para ajustar la altura del robot acorde a lo lleno o vacío que venga el saco.

En la imagen adjunta vemos un palet formado por **7 alturas** de sacos. Vemos que el espesor del saco son **190 mm**, por lo tanto la suma de las siete alturas más el valor del palet hace un total de 1475 mm de altura. Si tuviéramos en el ejemplo 10 alturas, la suma total sería de 2045mm. A esta altura y por las medidas de la instalación el robot puede llegar a tener acceso lo largo de todo el palet en todos los puntos de dejada sin problemas. El robot a pesar de estar subido en un soporte de acero cúbico, no se recomienda superar la altura máxima de 2500 mm, por lo que se restringe por programa el número de alturas máximas permitidas para paletizar. Hay que recordar que para que el robot deje el último saco de la última altura, este debe posicionarse por encima del saco por lo que la altura que precisa alcanzar el robot siempre es mayor. La limitación del número de

alturas no implica ningún problema puesto que rara vez por no decir nunca, se apilan más de 10 alturas de sacos en un palet. En caso de sobreexceder la limitación software de los ejes del robot, se procedería con las indicaciones dadas en capítulos sucesivos.

- **DIBUJO o MOSAICO.** El mosaico de un palet es la figura que queda representada en el palet al depositar todos los sacos de una altura, de esta manera si miramos un palet desde arriba veremos la figura que queda después de llenar una altura (Fig.6.1-2).



(Fig.6.1-2) Mosaico de 5 sacos

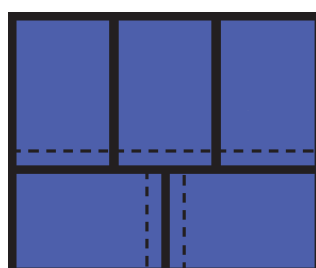
(Fig.6.1-3) Mosaico de 4 sacos

(Fig.6.1-4) Mosaico de 3 sacos

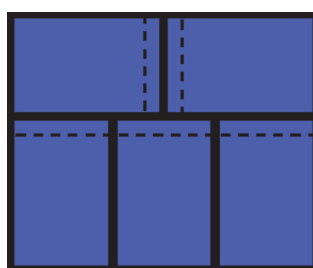


- **CAPA.** Para sacar una similitud, podemos comparar una capa con el período de una señal, de esta manera si la frecuencia de una señal esta relacionada con el número de veces que se repite un período, el número de capas de un mosaico está definida por el número de alturas necesarias que se necesitan apilar con el mosaico para conformar el palet, de manera que estas alturas se van alternando

o repitiendo hasta completar las alturas configuradas en el palet. Si tomamos como ejemplo un palet con mosaico de 5 sacos veremos que este palet está formado por dos capas diferentes, a partir de las cuales se repiten el resto.



(Fig.6.1-5) Mosaico de 5 CAPA-1



(Fig.6.1-6) Mosaico de 5 CAPA-2

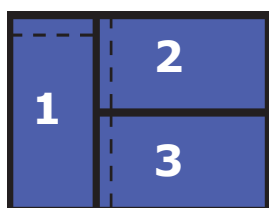
En las capas 1 y 2 lo único que varía es la disposición de los sacos. Es necesario ir alternando entre capa-1 y capa-2 para conformar un paquete compacto y cuadrado, ya que de lo contrario el palet sería uniforme.

- **NUMERO DE SACO O POSICIÓN.** Para realizar la paletización, el robot no utiliza el número de capas, utiliza el número de sacos. Así si nosotros configuramos por programa el mosaico_5 con 6 alturas, el robot paletizará 30 sacos. Estos sacos tienen una posición en el palet que irían del 1 al 30. El

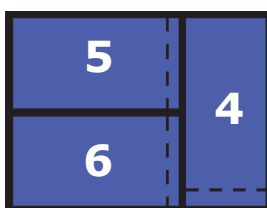
por qué paletizar por posición de saco es muy sencillo, y es que al ir alternando las capas que se repiten, en el caso que estamos comentando el saco nº11 de la altura 3 tendría la misma posición que el saco nº1 de la altura 1, de manera que las coordenadas X-Y de los sacos son las mismas desde el momento que se empiezan a repetir las capas. Solamente variará la altura Z, que será el incremento establecido por programa.

6.1.1 Configuración de tres sacos.

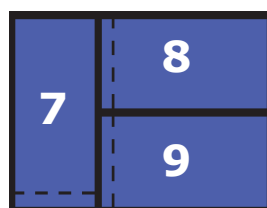
El mosaico de tres sacos, está formado por cuatro capas a partir de las cuales se van repitiendo. Son las siguientes:



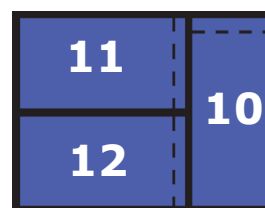
(Fig.6.1-7) Capa-1
Altura-1



(Fig.6.1-8) Capa-2
Altura-2

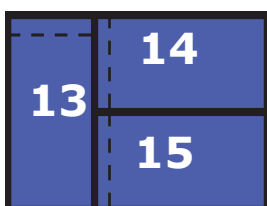


(Fig.6.1-9) Capa-3
Altura-3

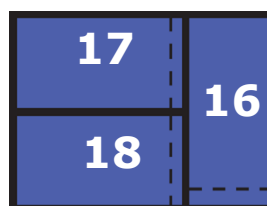


(Fig.6.1-10) Capa-4
Altura-4

De las cuatro capas diferentes los sacos 2-3-8-9 tienen las mismas coordenadas X-Y, lo mismo ocurre con los sacos 5-6-11-12. El único saco que tiene coordenadas diferentes en las cuatro capas son el 1-4-7-10; es decir las capas **Capa-1** y **Capa-3** son las mismas a excepción de los sacos 1-7 a los que se les da la vuelta a la boca para que el palet quede lo más estable y simétrico posible. Lo mismo ocurre con las capas **Capa-2** y **Capa-4** al que sólo se le da la vuelta a los sacos 4-10. A partir de estas cuatro capas para estas cuatro alturas, la quinta altura sería exactamente igual que la primera (saco-13 = saco-1, saco-14 = saco 2, saco-15 = saco-3), y la sexta altura igual que la segunda (saco-16 = saco-4, saco-17 = saco-5 y saco-18 = saco-6), así sucesivamente.



(Fig.6.1-11) Capa-1 Altura-5





(Fig.6.1-12) Capa-2 Altura-6

Para la programación de los puntos de dejada del robot Kuka, se tiene en cuenta como hemos mencionado anteriormente la posición del saco, así si observamos brevemente la programación del punto de dejada para el saco-1 veremos lo siguiente:

```

;;;saco 1 capa 1;;;;;;;;;;;;;
if (saco_1==1) or (saco_1==13) or (saco_1==25) then
PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

```

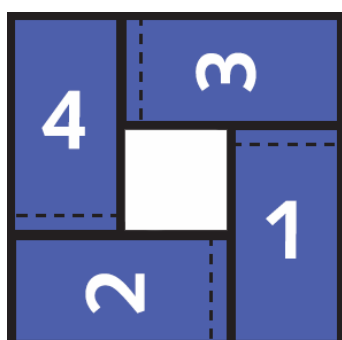
	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

En este ejemplo de subprograma **Zona_111 ()** del programa MOSAICO_3(), vemos que si la variable saco; la cual se irá incrementando mediante programa cada vez que se deposite un saco, tiene el valor de posición 1,13 ó 25, el punto de dejada será PTP P1, que es un punto referenciado a una base en la que solamente cambiará la coordenada Z el incremento de altura establecido por programa. Seguidamente mediante **pulse (\$out[19],true,2)** se activará durante dos segundos la electroválvula para la apertura de los dedos para dejar caer el saco.

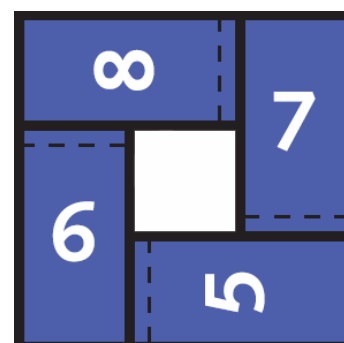
En este caso bastará con programar tres posiciones por saco (**saco_1==1**) or (**saco_1==13**) or (**saco_1==25**) ya que con tres posiciones por saco podríamos paletizar hasta 12 alturas. Recordar que en el mosaico de tres sacos, cada saco se repite cada 4 capas ó alturas, y el cliente como máximo sólo tendrá opción de paletizar hasta 10 alturas, por lo que programando tres posiciones por saco será suficiente.

6.1.2 Configuración de cuatro sacos.

Para paletizar un palet con un dibujo de cuatro sacos solamente necesitamos dos capas diferentes a partir de las cuales formar el resto del palet. Estas son las siguientes.



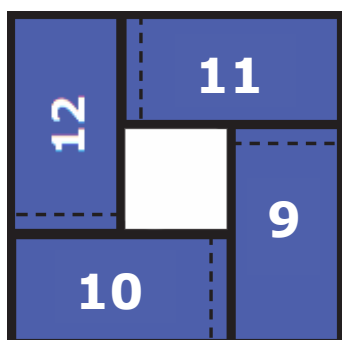
(Fig.6.1-13) Capa-1
Altura-1



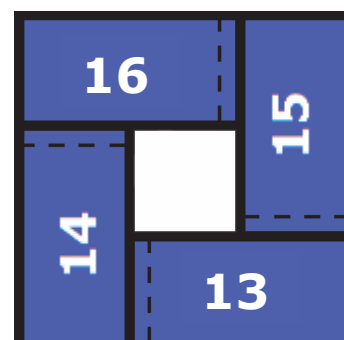
(Fig.6.1-14) Capa-2
Altura-2

Este tipo de mosaico, se paletiza en europalet de 1000*1200, sin embargo no se utilizará toda la superficie del palet puesto que la forma que obtendremos con este dibujo es un cuadrado con una especie de chimenea en su interior. Con dos capas será suficiente para configurar un palet compacto y homogéneo.



Las ocho posiciones de los ocho sacos que forman las dos capas, son diferentes, y se empezarán a repetir a partir de la tercera altura.



(Fig.6.1-15) Capa-1
Altura-3



(Fig.6.1-16) Capa-2
Altura-4

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

En la figura en cuanto a coordenadas X-Y se refiere, el punto de dejada del saco-9 = saco-1, saco-10 = saco-2, saco-11 = saco-3, saco-12 = saco-4, saco-13 = saco-5, saco-14 = saco-6, saco-15 = saco-7 y saco-16 = saco-8.

Tomaremos como ejemplo el subprograma **Zona_11 ()** del programa MOSAICO_4 ().

```

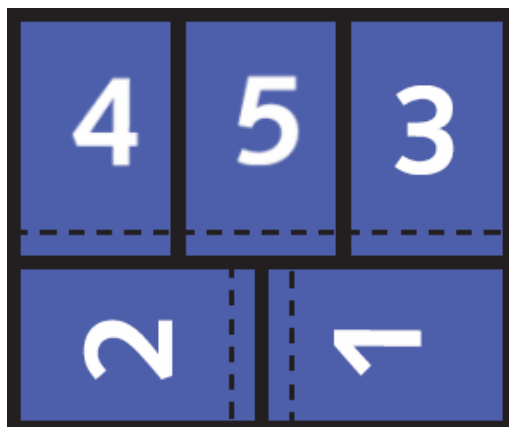
///////////////// saco 1 capa 1 ///////////////////
if (saco_1==1) or (saco_1==9) or (saco_1==17) or (saco_1==25) or (saco_1==33) or
(saco_1==41) or (saco_1==49) or (saco_1==57) then
PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

```

Todos los sacos con posición saco-1,9,17,25,33,41,49,57 tienen como coordenadas X-Y las del punto PTP P1, correspondiente al saco-1 de la capa-1. Al tener restringido el número de alturas en 10, nos bastaría con definir 5 posiciones por saco saco-1,9,17,25,33, no siendo necesario definir las posiciones saco_1 == 41, saco_1 == 49 y saco_1 == 57, ya que cada dos capas se repetirá.

6.1.3 Configuración de cinco sacos.

Al igual que la configuración anterior dos capas son necesarias para construir un palet con mosaico de 5 sacos por altura.



(Fig.6.1-16) Capa-1
Altura-1



(Fig.6.1-17) Capa-2
Altura-2

Con este dibujo abarcaremos toda la superficie del palet de 1000*1200. La filosofía de colocación es la misma que para el caso anterior, siendo la posición de los sacos en la altura 3, la misma que en la altura 1.

6.2 UN PROGRAMA DE ROBOT.

Se comenzará detallando la función de cada una de las variables declaradas en el programa, existen dos subcategorías:

1. **Variables Físicas.** En este apartado se engloba el conjunto de señales cableadas mediante el módulo **Beckhoff**, serán señales necesarias tanto para la comunicación con el autómata como para la toma de decisión mediante software de los componentes del robot.
2. **Variables Declaradas.** Se engloban en este apartado el conjunto de señales declaradas mediante programa para el manejo y flujo de control de programa y toma de decisiones.

6.2.1 Variables Físicas.

6.2.1.1 Entradas.

El Primer modulo de 8 bit's del beckhoff, está a destinado a señales de comunicación con el PLC mediante la configuración por **automático externo**, tal como se detalló en el capítulo anterior.

- **DI1-1 -> PGNO_BIT_1 -> \$IN[1] -> Entrada directa del Robot.**
- **DI1-2 -> PGNO_BIT_2 -> \$IN[2] -> Entrada directa del Robot.**
- **DI1-3 -> PGNO_BIT_3 -> \$IN[3] -> Entrada directa del Robot.**

Los bits de entrada 1,2 y 3 determinarán mediante combinación lógica el número de programa del robot a paletizar. Esta combinación lógica será enviada mediante salidas del PLC a dichas entradas del robot y guardada en una señal declarada en el programa de tipo signal la cual almacenará el valor lógico. De esta manera existen tres posibles combinaciones:

- PROGRAM=1 Programa de paletizado con mosaico de 5 sacos	{ \$IN[1] = true \$IN[2] = false \$IN[3] = false
- PROGRAM=2 Programa de paletizado con mosaico de 4 sacos	{ \$IN[1] = false \$IN[2] = true \$IN[3] = false
- PROGRAM=3 Programa de paletizado con mosaico de 3 sacos	{ \$IN[1] = true \$IN[2] = true \$IN[3] = false

Como puede verse el tercer bit de programa carece de uso, pero puede usarse como bit de paridad o como entrada reserva para futuras expansiones.

Como se ha venido diciendo en capítulos anteriores, el cambio de programa está condicionado a que la variable **PROGRAM** sea distinta al que se está ejecutando, y que los palet's y la zona de entrada de sacos estén vacíos.

```

;=====SALIDA CONDICIONAL=====
continue
if (((program==1) or (program==2)) and ($in[11]==false) and (saco_1<2) and (saco_2<2))
then
wait sec 1
exit
endif

endloop
wait sec 0
coi_sgrd ()

END
  
```

**FINAL DE PROGRAMA
MOSAICO_3 ()**

En el ejemplo que estemos paletizando con el programa **MOSAICO_3 ()**, tendremos que seleccionar desde el XBT un valor de programa distinto; es decir, **PROGRAM=1 or PROGRAM=2**, para poder salir del bucle y entrar nuevamente en el bucle de CELL, donde después de entrar en el programa seleccionado, tendremos que enviar los datos de altura.

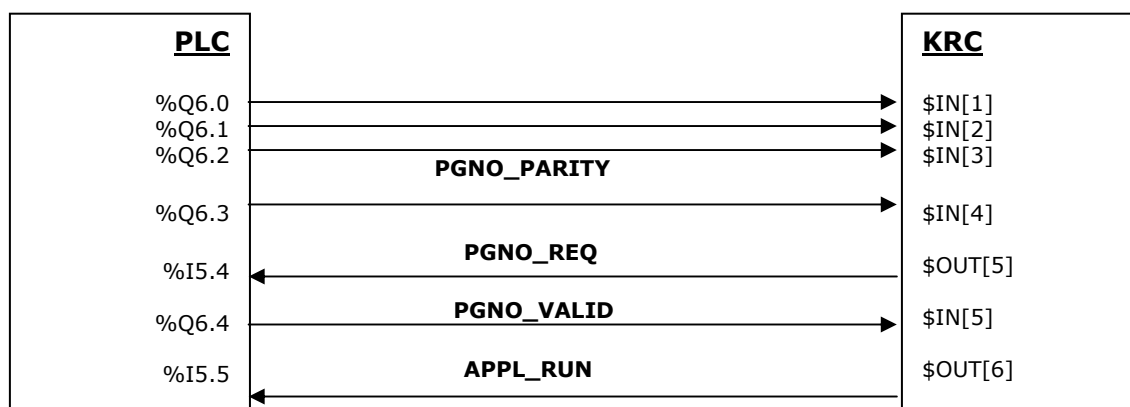
- **DI1-4 -> PGNO/PARITY -> \$IN[4] -> Entrada directa del Robot.**

El PLC mediante salida directa levantará esta señal hacia el robot para indicarle que se quiere transmitir un nuevo número de programa a través de las variables de entrada anteriores.



- **DI1-5 -> PGNO_VALID -> \$IN[5] -> Entrada directa del Robot.**

Como se explicó en el capítulo anterior, el PLC mediante salida directa le indicará al robot el permiso para realizar la lectura de la variable **PROGRAM**. Que recordemos guardará el número de programa seleccionado.

La Interface para el cambio de programa se resume así:



El PLC notifica mediante PGNO_PARITY que quiere enviar datos de programa. Mientras que el Robot responde con PGNO_REQ cuando se encuentre listo. Seguidamente el PLC escribe sobre las entradas de programa y lo confirma al

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

robot mediante la señal PGNO_VALID. Cuando se encuentre el programa de paletizado en RUN, el robot lo notificará al PLC mediante APPL_RUN.

- **DI1-6 -> EXT_ESTART -> \$IN[6] -> Entrada directa del Robot.**

Entrada del robot a través de la cual el PLC arrancará en RUN el programa **Cell.src** que será seleccionado la primera vez por el operador o cada vez que se reinicie el sistema. Esta acción será llevada a cabo siempre y cuando las seguridades estén activadas y los mensajes del robot confirmado.

- **DI1-7 -> MOVE_ENABLE -> \$IN[7] -> Entrada directa del Robot.**

Mediante esta entrada al robot el PLC, parará la ejecución de la línea de programa o la pondrá en marcha. Esta acción estará supeditada a la botonera del exterior del armario eléctrico donde se ubica el pulsador de paro.

- **DI1-8 -> CONF_MESS -> \$IN[8] -> Entrada directa del Robot.**

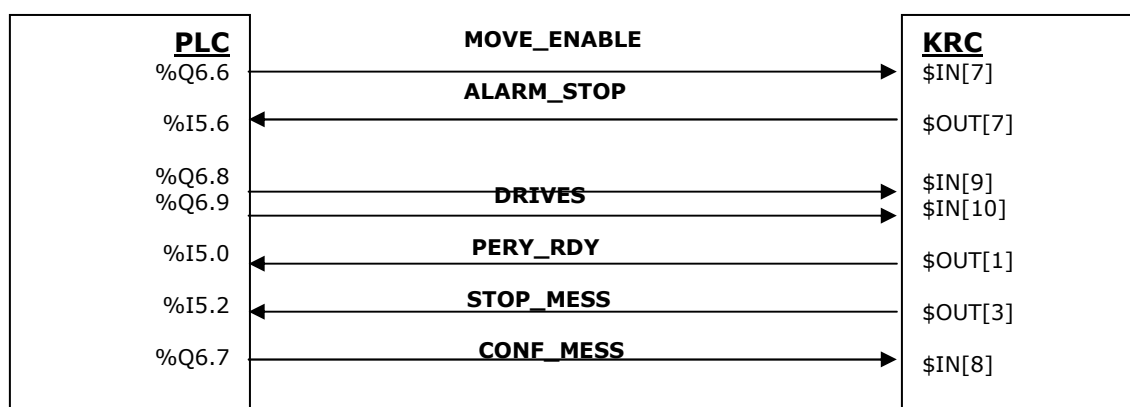
En la Interface del Automático externo, a través de esta entrada al robot el PLC confirmará los mensajes de seguridad, alarmas o advertencia de la ventana de mensajes, rearmándose el robot posteriormente bajo confirmación de este, si no existe ninguna anomalía.

El segundo módulo de entradas del Beckhoff, se establecen el resto de señales para la Interface del automático externo, y se definen las señales de permiso y descarga como veremos.



- **DI2-1 -> DRIVES_ON -> \$IN[9] -> Entrada directa del Robot.**
- **DI2-2 -> DRIVES_OFF -> \$IN[10] -> Entrada directa del Robot.**

En las entradas 9 y 10 del beckhoff el robot recibirá los estados de los accionamientos como si del KCP se tratara. Es preciso que **"DRIVES_OFF"** permanezca a nivel alto mientras el PLC manda un pulso a través de **"DRIVES_ON"** para activar los accionamientos del robot. De este modo el PLC tiene total control sobre los frenos del robot.

La Interface para el rearme del robot se resume de la siguiente manera:



El PLC tendrá que mantener la señal MOVE_ENABLE en estado alto, de lo contrario el robot no se pondrá en marcha. Si el KRC indica que no hay ninguna

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

seguridad activada mediante ALARM_STOP, el PLC pondrá los accionamientos al KRC manteniendo DRIVES_OFF en estado alto y enviando un pulso positivo por DRIVES_ON. Si los accionamientos han sido activados, el KRC lo notificará mediante PERI_RDY, así mismo notificará la existencia de mensajes por confirmar a través de STOP_MESS a lo que el PLC responderá mediante CONF_MESS.

- **DI2-3 -> Permiso_Carga_L1 -> \$IN[11] -> Entrada directa Robot.**



Terminadas las entradas de Interface automático externo. Mediante esta señal el PLC le dará consentimiento de carga al robot en los rodillos de alimentación de sacos; es decir, si la periferia está en marcha, un saco ha llegado al final de los rodillos de alimentación y el robot una vez haya cogido el saco tiene posibilidad de dejarlo; es decir, permiso de descarga en alguna de las líneas, entonces el PLC activará esta señal para indicarle al robot que puede recoger el saco.

- **DI2-4 -> Permiso_Carga_L2 (opcional) NO usada puesto que sólo existe una zona de recogida de saco.**
- **DI2-5 -> Permiso_Descarga_L1 -> \$IN[13] ->Entrada directa Robot.**
- **DI2-6 -> Permiso_Descarga_L2 -> \$IN[14] ->Entrada directa Robot.**

Las señales de permiso de descarga son indicadas por el PLC al robot cuando, estando la periferia en marcha, las líneas tienen activado el permiso de paletizar mediante los selectotes del armario eléctrico, y además el robot no indica al autómatas que el palet esta lleno mediante otra señal de palet finalizado. En estas circunstancias el PLC da permiso de descarga al robot en la línea 1 ó 2, indicando que puede dejar el saco que anteriormente recogió.

- **DI2-7 -> Carga_Datos_L1 -> \$IN[15] -> Entrada directa del Robot.**
- **DI2-8 -> Carga_Datos_L2 (opcional) NO usada puesto que los datos cargados en L2 son los mismos que en L1.**
- **DI3-1 -> GND (Señal False) -> \$IN[17] -> Entrada directa de Robot.**
- **DI3-2 -> Carga_Datos_Capas -> \$IN[18] -> Entrada directa Robot.**
- **DI3-3 -> Carga_Datos_Ordenes -> \$IN[19] ->Entrada directa Robot.**
- **DI3-4 -> Carga_Datos_Incr_Alturas -> \$IN[20] -> Entrada directa del Robot.**
- **DI3-5 -> DATO_1 -> \$IN[21] -> Entrada directa del Robot.**
- **DI3-6 -> DATO_2 -> \$IN[22] -> Entrada directa del Robot.**
- **DI3-7 -> DATO_3 -> \$IN[23] -> Entrada directa del Robot.**
- **DI3-8 -> DATO_4 -> \$IN[24] -> Entrada directa del Robot.**

Las señales de carga de datos direccionadas desde la **\$IN[15]** hasta la **\$IN[20]**, siguen un estándar de comunicación a partir del las cuales el autómatas indica al programa del robot la carga de datos o instrucciones (Nº de mosaico, Nº de capas e Incremento de alturas) y (Mando ir a HOME, Mantenimiento o Liberación) desde el terminal de diálogo XBT. Al decir estándar significa que por ejemplo **\$IN[16]** estará destinada a **reserva de datos de carga** en la LINEA_2 ya que comparte los mismos datos de carga que la LINEA_1. Si se usará en el programa **\$IN[17]** pero estará en estado **"false"** también en previsión de futuras expansiones de datos. De esta manera según el tipo de dato que se quiera cargar el autómatas activará las siguientes combinaciones de señales:

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

- **ORDEN HOME.
MANTENIMIENTO
LIBERACIÓN** { **\$IN[15] = No usada.
\$IN[17] = false
\$IN[18] = false
\$IN[19] = true
\$IN[20] = false**

Si desde el terminal XBT ejecutamos la orden de ir a HOME, MANTENIMIENTO o LIBERACIÓN, el PLC activa la señal **\$IN[19]** indicando al robot que queremos ejecutar una orden. Dependiendo de la orden, se transferirá a través de los datos de programa un valor que determinará la acción. Estos valores que envía el PLC a través de **DATO_1, DATO_2, DATO_3 y DATO_4**, están declarados en el robot como entradas; al igual que ocurría con la variable **PROGRAM**, mediante una variable de tipo signal denominada **registro_1**, de manera que la combinación binaria de estas señales será el número a transferir y guardado en dicha variable.

Así por ejemplo tendremos las siguientes opciones:

	\$IN[24]->DATO_4	\$IN[23]->DATO_3	\$IN[22]->DATO_2	\$IN[21]->DATO_1
Registro_1 = 1 HOME	false	false	false	true
Registro_1 = 2 MANTENIMIENTO	false	false	true	false
Registro_1 = 3 LIBERACION	false	false	true	true

Tabla: ORDENES del Robot.

Si desde el terminal XBT, seleccionamos HOME, la variable registro_1 tomará el valor 1, si seleccionamos MANTENIMIENTO, la variable tomará el valor 2 y si LIBERAMOS, la variable tomará el valor 3. De esta manera si estamos ejecutando un programa de paletizado la variable registro será una condición directa para entrar en los subprogramas **Home ()** o **Mantenimiento ()** a parte de otras condiciones lógicas.

<pre>=====MANTENIMIENTO===== continue if ((\$in[17]==false) and (\$in[18]==false) and(\$in[19]==true) and (\$in[20]==false) and (registro_1==2) then mantenimiento() endif =====ZONA PALETIZADO 1===== continue if \$out[25]==false then continue if ((\$in[11]==true) and (\$in[13]==true) and (\$out[30]==false)) then zona_11 () endif endif =====ZONA PALETIZADO 2===== continue if \$out[25]==false then continue</pre>	<div>DENTRO DEL BUCLE PROGRAMA MOSAICO_3 ()</div>
--	---

```

if (($in[11]==true) and ($in[14]==true) and ($out[30]==true)) then
zona_22 ()
endif
endif
;=====HOME=====
continue
if (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false) and
(registro_1==1) and (saco_1<2) and (saco_2<2)) then
home ()
endif

```

- **CARGA DATOS DE CAPAS.**

{

 \$IN[15] = true

 \$IN[17] = false

 \$IN[18] = true

 \$IN[19] = false

 \$IN[20] = false

Si el autómata activa esta combinación de señales indicará al programa robot que el dato a transferir desde EL TERMINAL XBT y a través del autómata será el número de capas (alturas). Mediante los datos **DATO_1, DATO_2, DATO_3** y **DATO_4** transferiremos el número de alturas a paletizar que será guardado en la variable de tipo signal denominada **registro_1**. Para el caso de arriba en el que se quiera transferir el número de capas, las señales de salida del PLC de tipo dato con entrada directa al autómata tomarán los siguientes valores.

	\$IN[24]->DATO_4	\$IN[23]->DATO_3	\$IN[22]->DATO_2	\$IN[21]->DATO_1
Registro_1 = 1 NºAlturas = 1	false	false	false	true
Registro_1 = 2 NºAlturas = 2	false	false	true	false
Registro_1 = 3 NºAlturas = 3	false	false	true	true
Registro_1 = 4 NºAlturas = 4	false	true	false	false
Registro_1 = 5 NºAlturas = 5	false	true	false	true
Registro_1 = 6 NºAlturas = 6	false	true	true	false
Registro_1 = 7 NºAlturas = 7	false	true	true	true
Registro_1 = 8 NºAlturas = 8	true	false	false	false
Registro_1 = 9 NºAlturas = 9	true	false	false	true
Registro_1 = 10 NºAlturas = 10	true	false	true	false

Tabla: N° de alturas del Robot.

En el submit o programa multitarea, se guarda constantemente el valor de la variable declarada **registro_1** en otra de nombre **reg_capas_1**.

```

;=====CAPTURA DE DATOS BINARIOS=====
;;;;;;;;;;;;;Capas
if (($in[15]==true) and ($in[17]==false) and ($in[18]==true) and ($in[19]==false) and
($in[20]==false)) then
  reg_capas_1=registro_1
endif
  
```

De manera que cuando el robot se encuentre en HOME; indicado al PLC por medio de otra variable de programa denominada **\$OUT[21] = true**, el programa de paletizado permitirá la captura del número de alturas en la variable **capas_1** para el cálculo de los sacos totales a paletizar. Esto será después de que el robot sea liberado desde las opciones del XBT. De esta manera podremos hacer los cambios que queramos sin que el robot se mueva

```

DEF home( )
coi_sgrd ( )

;marca para saber que estamos dentro del Home y cambiar de programa
$out[21]=true

wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false)
and (registro_1==3))

;=====CAMBIO DE MOSAICOS=====

;Mosaico seleccionado
$out[21]=false

;=====CAMBIO DE CAPAS=====

capas_1=reg_capas_1
if program==1 then
  sacos_1=capas_1*5
  sacos_2=capas_1*5
endif
if program==2 then
  sacos_1=capas_1*4
  sacos_2=capas_1*4
endif
if program==3 then
  sacos_1=capas_1*3
  sacos_2=capas_1*3
endif

END
  
```

SUBPROGRAMA HOME ()

Cumplidas todas las condiciones lógicas necesarias para entrar en Home(), el programa esperará la condición necesaria para su **liberación**. Mientras no liberemos y estemos en home(), transferiremos desde el XBT el número de alturas, para que en función del programa seleccionado se calculen los sacos totales a guardar en la variable declarada **sacos_1** y **sacos_2** para cada una de las líneas.

- **CARGA DATOS INCREMENTOS.**

{

 \$IN[15] = true

 \$IN[17] = false

 \$IN[18] = false

 \$IN[19] = false

 \$IN[20] = true



 }

Con la señal **\$IN[20]** puesta a true, el PLC indica al robot por medio del programa del submit, que el valor a enviar a través de las variables de dato será el incremento de altura. En este caso se utilizará la señal "**registro_2**" también declarada como signal. Se usarán las variables DATO_1, DATO_2 y DATO_3 como valor a incrementar y DATO_4 como bit de signo.

	\$IN[24]->DATO_4	\$IN[23]->DATO_3	\$IN[22]->DATO_2	\$IN[21]->DATO_1	$\Delta \pm$ (mm)
Registro_2 = 0 Incr.Alt = 0		false	false	false	0
Registro_2 = 1 Incr.Alt = 1	false	false	false	true	(+)1
Registro_2 = 2 Incr.Alt = 2	false	false	true	false	(+)2
Registro_2 = 3 Incr.Alt = 3	false	false	true	true	(+)3
Registro_2 = 4 Incr.Alt = 4	false	true	false	false	(+)4
Registro_2 = 5 Incr.Alt = 5	false	true	false	true	(+)5
Registro_2 = 6 Incr.Alt = 6	false	true	true	false	(+)6
Registro_2 = 7 Incr.Alt = 7	false	true	true	true	(+)7
Registro_2 = -1 Incr.Alt = -1	true	false	false	true	(-)1
Registro_2 = -2 Incr.Alt = -2	true	false	true	false	(-)2
Registro_2 = -3 Incr.Alt = -3	true	false	true	true	(-)3
Registro_2 = -4 Incr.Alt = -4	true	true	false	false	(-)4
Registro_2 = -5 Incr.Alt = -5	true	true	false	true	(-)5
Registro_2 = -6 Incr.Alt = -6	true	true	true	false	(-)6
Registro_2 = -7 Incr.Alt = -7	true	true	true	true	(-)7

Tabla: Incrementos de altura del robot.

En el programa multitarea, este valor es cargado y guardado en la variable "**altura_1**" que será empleada para incrementar el valor de la coordenada Z en cada nueva altura multiplicado por cinco, de manera que cada incremento equivale a 5 mm.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

//////////Incrementos

```

if (($in[15]==true) and ($in[17]==false) and ($in[18]==false) and ($in[19]==false) and
($in[20]==true)) then
if $in[24]==false then
reg_alt_incr_1=registro_2
altura_1=reg_alt_incr_1*5
endif
if $in[24]==true then
reg_alt_incr_1=0-registro_2
altura_1=reg_alt_incr_1*5
endif
endif

```

**PROGRAMA
MULTITAREA**

- **DI4-1 -> Restos -> \$IN[25] -> Entrada directa del Robot.**

La señal de restos, no tiene ninguna conexión con el PLC, siendo esta entrada de robot accionada directamente desde el armario eléctrico y a través del pulsador para tal fin. Mediante esta entrada el robot recibirá la orden de finalizar cada palet de cada una de las líneas. Comunicándose de esta manera al PLC, de manera que una vez terminados los movimientos en ejecución el robot se detendrá en espera de que los palet's sean sustituidos.

Desde el programa multitarea se captura esta pulsación, poniendo a true la variable \$OUT[25]. En el momento que esta variable se active, se deshabilitará el paletizado en ambas líneas y el PLC quitará el permiso de carga y descarga hasta que el operario coloque un palet nuevo.

;=====RESTOS=====

```

continue
if $out[25]==true then
wait sec 1
pulse ($out[10],true,2)
pulse ($out[11],true,2)
saco_1=1
saco_2=1
base_data[3]=base_data[1]
base_data[4]=base_data[2]
wait sec 3
$out[25]=false
endif



```

**DENTRO DEL BUCLE
PROGRAMA MOSAICO_3 ()**

En el momento de producirse el restos, y después de que el robot salga de las zonas de paletizado correspondiente. Se envía al PLC un pulso de dos segundos en las señales de fin de palet para que este quite los permisos de carga y descarga de las líneas, seguidamente se resetean las variables de paletizado **saco_1** y **saco_2** y se reinician las bases. Para que el próximo palet lo empiece desde la altura inicial.

- **DI4-3 -> Manos_abiertas -> \$IN[27] -> Entrada directa del Robot.**
- **DI4-4 -> Dedos_abiertos -> \$IN[28] -> Entrada directa del Robot.**

Como se ha ido explicando a lo largo del proyecto, estas entradas indican al robot la posición de reposo de la mordaza una vez que ha dejado el saco en el palet. De esta manera transcurrido un tiempo; y en caso de no adquirir la posición

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

de reposo, enviaremos una señal de alarma al PLC, que será transmitida por pantalla. (Ver capítulos de Alarmas XBT.)

6.2.1.2 Salidas.

- **DO1-1 -> PERY_RDY -> \$OUT[1] -> Salida directa del Robot.**
- **DO1-2 -> I_O_ACTCONF -> \$OUT[2] -> Salida directa del Robot**
- **DO1-3 -> STOP_MESS -> \$OUT[3] -> Salida directa del Robot**
- **DO1-4 -> PRO_ACT -> \$OUT[4] -> Salida directa del Robot**
- **DO1-5 -> PGNO_REQ -> \$OUT[5] -> Salida directa del Robot**
- **DO1-6 -> APPL_RUN -> \$OUT[6] -> Salida directa del Robot**
- **DO1-7 -> ALARM_STOP -> \$OUT[7] -> Salida directa del Robot**
- **DO1-8 -> MODO_T1 -> \$OUT[8] -> Salida directa del Robot**

El primer byte de salidas del bechhoff, está destinado a la Interface de salida para el manejo mediante automático externo desde el PLC (Ver capítulo Automático externo). En esta Interface se emplea la señal **PERY_RDY** para indicar al PLC, que el robot tiene los accionamientos desconectados, **I_O_ACTCONF** para indicar que los drivers de comunicación del robot se encuentran reestablecidos en la tarjeta electrónica, **STOP_MESS** que indicará al PLC la existencia de mensajes por confirmar, **PRO_ACT** señal activada siempre y cuando haya algún proceso activado en la unidad del KRC (cuando se inicia Cell.src), **PGNO_REQ** donde el KRC da consentimiento al PLC para el envío de un nuevo programa, **APPL_RUN** para identificar que algún programa de paletizado se encuentra activo y **ALARM_STOP** para identificar al PLC la existencia de una emergencia que tendrá que ser solventada y posteriormente confirmada.



Ha estas señales el robot tiene configurada otras dos salidas que son la identificación del modo de funcionamiento para la elección de procesos desde el PLC. Así pues **\$OUT[8]** el KRC indicará a la unidad de control si el selector de operaciones se encuentra en manual o no, ya que de estar el robot en manual no podríamos activar los accionamientos desde el PLC, e inmediatamente se deshabilitaría la señal **MOVE_ENABLE** provocando cualquier paro de movimiento automático en el KRC.

- **DO2-1 -> MODO_EXTERN -> \$OUT[9] -> Salida directa del Robot.**

Mediante **\$OUT[9]** el KRC indica al PLC su estado de automático y control mediante medios externos. Su estado activado mediante el selector será necesario para poner en marcha el robot y habilitar los movimientos.

- **DO2-2 -> Fin_Palet_L1 -> \$OUT[10] -> Salida directa del Robot.**
- **DO2-3 -> Fin_Palet_L2 -> \$OUT[11] -> Salida directa del Robot.**

Las señales de fin de palet, le indican al PLC, que se ha finalizado un palet en cada una de las líneas y que por lo tanto se precisa cambiarlo mediante el proceso habitual desde el armario eléctrico inhibiendo las barreras y generando un nuevo palet tal como se vio en capítulos anteriores. Durante todo este proceso el robot permanecerá detenido esperando nueva orden, ya que una vez que el KRC notifica al PLC la finalización de un palet, este inmediatamente le quita el consentimiento de carga y descarga, por lo que el robot no entra en ninguna zona de paletizado.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

La finalización del palet se puede dar en dos casos.

1. Se ha pulsado el botón de restos, en cuyo caso se finaliza en ambas líneas tal como se vio.
2. El proceso ha finalizado por alcance del número máximo de alturas y de sacos configurado desde el XBT.

La finalización por alcance del número máximo de alturas y de sacos, se da dentro de los subprogramas de zona denominados "**Zona_1 ()**" para el mosaico de 5 sacos en la línea 1, "**Zona_11 ()**" para el mosaico de 4 sacos en la línea 1, "**Zona_111 ()**" para el mosaico de 3 sacos en la línea 1, "**Zona_2 ()**" para el mosaico de 5 sacos en la línea 2, "**Zona_22 ()**" para el mosaico de 4 sacos en la línea 2 y "**Zona_222 ()**" para el mosaico de 3 sacos en la línea 2. Así pues si analizamos una parte del código final de los subprogramas de paletizado tenemos los siguientes:

```

;;;;;;;;permiso para sacar palet;;;;;;;;
wait sec 0
if (saco_1==sacos_1) then
pulse ($out[10],true,2)
saco_1=0
base_data[3]=base_data[3]
wait sec 0.5
endif

;;;;;;;;incremento saco;;;;;;;;

saco_1=saco_1+1

```



**DENTRO Subprograma
Zona ()**

Después de depositar el saco dentro del subprograma zona (), se comprueba si la variable de posición del saco "**saco_1**" o "**saco_2**" (para la línea 2), es igual al número de sacos máximos a paletizar calculado dentro del **home ()** y guardado en las variables "**sacos_1**" o "**sacos_2**". Si es así se envía al PLC el pulso de final del palet en esa línea y se reinicia la variable de posición del saco así como la base que ha ido incrementando en cada altura. Todo ello igual que ocurría con el resto. En caso de que la igualdad no se cumpla, se incrementa la variable de posición **saco_1 = saco_1 + 1** de manera que cuando volvamos a entrar en la zona para dejar el saco correspondiente, volvamos a verificar esta igualdad pero con la variable incrementada en una unidad.

- **DO2-4 -> Defecto_Mordaza -> \$OUT[12] -> Salida directa del Robot.**

El defecto de mordaza, es proporcionado por el KRC, una vez que el robot no alcanza la posición de reposo de la mordaza transcurrido 4 seg de reloj desde que el robot alcanza la posición de reposo en la mesa de recogida de entrada de sacos. (Ver Alarmas. Capítulo del XBT).

- **DO3-1 -> Abrir_Manos -> \$OUT[17] -> Salida directa del Robot.**
- **DO3-1 -> Cerrar_Manos -> \$OUT[18] -> Salida directa del Robot.**
- **DO3-1 -> Abrir_Dedos -> \$OUT[19] -> Salida directa del Robot.**
- **DO3-1 -> Cerrar_Dedos -> \$OUT[20] -> Salida directa del Robot.**

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

Mediante estas salidas directas del robot, la unidad controla los accionamientos de la mordaza. No tienen ninguna comunicación con el PLC y son gobernadas única y exclusivamente por el KRC. (**Ver Componentes Hardware de la Garra**).

- **DO3-1 -> Dentro_HOME -> \$OUT[21] -> Salida directa del Robot.**

\$OUT[21] será la señal que el KRC activará para indicar al PLC que se encuentra en el estado óptimo preparado para el cambio de programa y por tanto de las alturas; es decir, dentro del subprograma **home()** una vez se hayan producido todas las condiciones lógicas necesarias para ello. Esta señal permite la recepción del nuevo envío de programa desde el PLC a través de los bit's de programa **PGNO_BIT** y una vez liberado del home, la asignación de los sacos totales en función del programa seleccionado y del número de alturas transmitido. Podemos cambiar sólo el número de alturas por eso para entrar en home será condición indispensable que los palet's están vacíos. Una vez cambiado liberado el robot y cambiado el número de alturas, si el programa sigue siendo el mismo, continuará con el programa de paletizado en ejecución, pero si la variable **PROGRAM** ha cambiado, entonces saldremos del bucle del programa de paletizado para entrar nuevamente en **cell.src** donde se seleccionará el programa adecuado.



(Fig.6.2-1) Captura del valor de programa en HOME

En esta parte del PLC, se establece que si el KRC, está **dentro de home ()** se produce una igualdad entre las variables **%MW1** (Programa seleccionado desde el XBT) y **%MW18** (programa a visualizar también en el XBT) de manera que los tres primeros bit's de esta son transmitidos por las líneas PGNO_BIT que llegarán hasta el bechhoff del robot en forma de entradas y se modificará así la señal **PROGRAM**.

6.2.2 Variables declaradas.

En este apartado se definirán aquellas señales que sin disponer de ningún tipo de conexión eléctrica son declaradas como marcas de memoria para el control de los procesos internos de los programas y subprogramas. Todas ellas ya vienen siendo comentadas en capítulos anteriores y se definen en el archivo de configuración "**\$Config.Dat**". Son las siguientes:

```

;=====
; Userdefined Variables
;=====
;DECL INT CONTEO=0
SIGNAL registro_1 $in[21] TO $in[24]
SIGNAL registro_4 $in[21] TO $in[24]
SIGNAL registro_2 $in[21] TO $in[23]
SIGNAL program $in[1] TO $in[3]
DECL INT sacos_1=40
DECL INT sacos_2=40
;decl int mosaico_1=1
;decl int reg_mosaico_1=1
;decl int mosaico_2=1
;decl int reg_mosaico_2=1
DECL INT reg_capas_1=8
;decl int reg_capas_2=8
DECL INT alt_incr_1=180
DECL INT altura_1=0
;decl int altura_2=0
;decl int alt_incr_2=160
DECL INT reg_alt_incr_1=0
;decl int reg_alt_incr_2=0
DECL INT saco_1=1
DECL INT saco_2=1
DECL INT capas_1=8
;decl int capas_2=8
;decl int r_sacos_1=0
;decl int r_sacos_2=0
;DECL INT zvel=80
;DECL FRAME basepalet1
;FRAME base_2={x 0.0,y 0.0,z 0.0,a 0.0,b 0.0,c 0.0}

;ENDFOLD (USER GLOBALS)
ENDDAT

```

Declaración en
\$Config.dat



Como puede apreciarse, no todas las señales declaradas son empleadas, siendo aquellas variables precedidas del ";" usadas como comentarios sin utilidad alguna en esta aplicación.

Las variables de tipo Signal, como hemos venido comentando, son variables que en este caso adquieren el valor decimal de la combinación binaria de los bit's integrantes que los forman. De esta manera:

- **SIGNAL registro_1 \$in[21] TO \$in[24].** El valor que adquiera esta señal dependerá de los valores transmitidos por el PLC desde los bit's de salida DATO_1, DATO_2, DATO_3 y DATO_4 hacia estas cuatro entradas del robot. La variable **registro_1** guardará pues, la instrucción o la orden a cargar desde los bit's \$IN[15] hasta \$IN[20] así como el valor de alturas a paletizar y los incrementos de altura.

CARGA ORDENES \$IN[15] a \$IN[20]		CARGA DATOS alturas \$IN[15] a \$IN[20]	CARGA DATOS Incrementos \$IN[15] a \$IN[20]
registro_1 = 1	Orden ir a HOME		
registro_1 = 2	Orden ir a MANTENIMIENTO	registro_1 = Nº alturas reg_capas_1 = registro_1	registro_2 = Δ± alturas reg_alt_incr_1 = registro_2 altura_1 = reg_alt_incr_1*5
registro_1 = 3	Orden LIBERACION		

Tabla: registro_1 del Robot.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

- **SIGNAL registro_2 \$in[21] TO \$in[23].** En registro_2 se guardará el valor del incremento de altura que configuremos desde el terminal XBT, siendo DATO_4 -> \$IN[24] el bit de signo.
- **SIGNAL program \$in[1] TO \$in[3].** Valor de la variable %MW18 del PLC donde guardamos el número de programa a paletizar cuando estamos dentro del home ().
- **DECL INT reg_capas_1=8.** El valor reg_capas_1 está declarada como variable entera, pudiendo ser capaz de almacenar un número entero de hasta 32 bit's de resolución. Las variables declaradas como enteras deben ser inicializadas con un valor, en este caso se inicializa con el valor 8 representando así 8 alturas de paletizado, aunque este valor no es determinante ni funcional en el programa, adquiriendo en el programa el valor que le corresponda y siendo constantemente actualizado en el config.

La variable **reg_capas_1** será usada por el programa multitarea **SPS.sub** para almacenar el valor de las capas desde **registro_1** cada vez que modifiquemos el número de alturas desde el terminal XBT. De manera que una vez que estemos dentro del **home ()** usaremos este valor para el cálculo de los sacos totales.

```

//////////Capas
if (($in[15]==true) and ($in[17]==false) and ($in[18]==true) and ($in[19]==false) and
($in[20]==false)) then
reg_capas_1=registro_1
endif

```

Dentro del
SPS.sub

El valor de las alturas transmitido desde el terminal XBT, será guardado y capturado en esta variable en cualquier momento de ejecución del programa.

- **DECL INT capas_1=8**
- **DECL INT capas_2=8**

Declaradas también como variables enteras y con el valor 8 preasignado, estas variables determinarán las alturas de la línea 1 y las alturas para la línea 2 dentro del **home()**. El hecho de disponer de dos variables se debe a que hay instalaciones en las que tenemos la opción de que el número de alturas no sea el mismo en ambas líneas, no será el caso aquí en el que tanto capas_1 como capas_2 tendrán el mismo valor en el programa ya que ambas dependen de reg_capas_1.

- **DECL INT sacos_1=40**
- **DECL INT sacos_2=40**

Estas dos variables enteras, sirven para almacenar la cantidad de sacos totales a paletizar en cada una de las dos líneas en función del programa seleccionado. Será desde home (), donde en función de la variable PROGRAM se adquiera un valor u otro.

```

DEF home( )
coi_sgrd ()

;marca para saber que estamos dentro del Home y cambiar de programa
$out[21]=true

```



```
wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false)
and (registro_1==3))
;=====CAMBIO DE MOSAICOS=====
;Mosaico seleccionado
$out[21]=false
;=====CAMBIO DE CAPAS=====
capas_1=reg_capas_1
if program==1 then
sacos_1=capas_1*5
sacos_2=capas_1*5
endif
if program==2 then
sacos_1=capas_1*4
sacos_2=capas_1*4
endif
if program==3 then
sacos_1=capas_1*3
sacos_2=capas_1*3
endif
END
```

**SUBPROGRAMA
HOME ()**

Cuando hemos entrado en **home()** con las líneas totalmente vacías, y después de liberar y enviar los datos de programa y alturas a paletizar. En **capas_1** se guarda el valor de **reg_capas_1** que a su vez contiene el valor de **registro_1** cargado desde el submit cuando se enviaron las alturas desde el XBT. Después y dependiendo del programa seleccionado y enviado, la variable **sacos_1** y **sacos_2** adquirirá el mismo valor, siendo este el número de alturas por el número de sacos por altura en función del programas, así tendremos los siguientes valores para las variables **sacos_1** y **sacos_2**. (Véase la siguiente tabla.)

	PROGRAM = 1 Mosaico_5 ()	PROGRAM = 2 Mosaico_4 ()	PROGRAM = 3 Mosaico_3 ()
reg_capas_1 = 1 capas_1 = 1	sacos_1 = 5 sacos_2 = 5	sacos_1 = 4 sacos_2 = 4	sacos_1 = 3 sacos_2 = 3
reg_capas_1 = 2 capas_1 = 2	sacos_1 = 10 sacos_2 = 10	sacos_1 = 8 sacos_2 = 8	sacos_1 = 6 sacos_2 = 6
reg_capas_1 = 3 capas_1 = 3	sacos_1 = 15 sacos_2 = 15	sacos_1 = 12 sacos_2 = 12	sacos_1 = 9 sacos_2 = 9
reg_capas_1 = 4 capas_1 = 4	sacos_1 = 20 sacos_2 = 20	sacos_1 = 16 sacos_2 = 16	sacos_1 = 12 sacos_2 = 12
reg_capas_1 = 5 capas_1 = 5	sacos_1 = 25 sacos_2 = 25	sacos_1 = 20 sacos_2 = 20	sacos_1 = 15 sacos_2 = 15
reg_capas_1 = 6 capas_1 = 6	sacos_1 = 30 sacos_2 = 30	sacos_1 = 24 sacos_2 = 24	sacos_1 = 18 sacos_2 = 18
reg_capas_1 = 7 capas_1 = 7	sacos_1 = 35 sacos_2 = 35	sacos_1 = 28 sacos_2 = 28	sacos_1 = 21 sacos_2 = 21
reg_capas_1 = 8 capas_1 = 8	sacos_1 = 40 sacos_2 = 40	sacos_1 = 32 sacos_2 = 32	sacos_1 = 24 sacos_2 = 24
reg_capas_1 = 9 capas_1 = 9	sacos_1 = 45 sacos_2 = 45	sacos_1 = 36 sacos_2 = 36	sacos_1 = 27 sacos_2 = 27
reg_capas_1 = 10 capas_1 = 10	sacos_1 = 50 sacos_2 = 50	sacos_1 = 40 sacos_2 = 40	sacos_1 = 30 sacos_2 = 30
	sacos_1 = capas_1*5 sacos_2 = capas_1*5	sacos_1 = capas_1*4 sacos_2 = capas_1*4	sacos_1 = capas_1*3 sacos_2 = capas_1*3

Tabla: N° de sacos del Robot.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

- **DECL INT saco_1=1**
- **DECL INT saco_2=1**

Las variables **saco_1** y **saco_2** inicializadas con el valor "1" determinan la posición del saco en el palet en función de la cantidad de sacos totales; es decir, **saco_1** y **saco_2** se irá incrementando por cada saco colocado hasta alcanzar el valor máximo, donde el KRC indicará al PLC que el palet ha sido finalizado mediante la señal correspondiente.

- **DECL INT alt_incr_1=180**
- **DECL INT reg_alt_incr_1 = 0**
- **DECL INT altura_1=0**

Estas tres variables enteras, se encuentran directamente relacionadas en el programa multitarea submit y en los subprogramas de zona para incrementar las bases de paletizado, mientras en la variable **reg_alt_incr_1** se almacena el valor de incremento de **registro_2**, en la variable **altura_1** se determina el valor total a incrementar multiplicado por 5, tanto para la línea 1 como para la línea 2. La variable **alt_incr_1** se utiliza como valor constante de incremento dentro del programa (Véase ejemplo en **Base_data []**).

6.2.2.1 Base_data [].

En el archivo de configuración también se declaran las bases de trabajo.

Podemos definir hasta 32 bases diferentes según queramos dar uso. Esto será necesario en nuestro caso para que las coordenadas de los puntos queden perpendiculares y paralelas a las coordenadas X-Y de desplazamiento del robot, de esta manera; y siempre y cuando la orientación de la garra en cada punto ya se encuentre ajustada, podremos editar los archivos de coordenadas y modificar la posición de los sacos teniendo en cuenta estas coordenadas.

Se definen **BASE_DATA[1]** y **BASE_DATA[3]** para la línea 1 y **BASE_DATA[2]** y **BASE_DATA[4]** para la línea 2 con los valores mostrados a continuación.

```
INT MAX_BASE=32
DECL FRAME BASE_DATA[32]
BASE_DATA[1]={x -1431.70801,y 1425.22803,z 75.8300018,a -40.6877632,b -
0.00680800015,c 3.99999999E-05}
BASE_DATA[2]={x -1382.01294,y -1494.18701,z 88.7521362,a 47.8801079,b -
0.000916999998,c 0.00118899997}
BASE_DATA[3]={x -1431.70801,y 1425.22803,z 75.8300018,a -40.6877632,b -
0.00680800015,c 3.99999999E-05}
BASE_DATA[4]={x -1382.01294,y -1494.18701,z 88.7521362,a 47.8801079,b -
0.000916999998,c 0.00118899997}
BASE_DATA[5]={x 0.0,y 0.0,z 0.0,a 0.0,b 0.0,c 0.0}
```

BASE_DATA[1] y **BASE_DATA[2]** serán las bases fijas, mientras que **BASE_DATA[3]** y **BASE_DATA[4]** serán las bases a las que iremos incrementando el valor de la coordenada Z por cada altura del palet terminada, de manera que una vez finalizado el palet, podamos igualar las bases con sus valores originales para comenzar de nuevo desde la altura nº 1. Sírvese como ejemplo el siguiente trozo de uno de los subprogramas de zona de paletizado.

DEF zona_1()

```

////////////////////////////////////
#####  #  #  #####
#  #  #  #  #
#  #  #  #  #
#####  #  #  #
////////////////////////////////////

```

```

alt_incr_1=190
$out[30]=true
coger_1 ( )

```

**INICIO
SUBPROGRAMA
ZONA ()**

```

;;;saco 1 capa 1;

```

```

if (saco_1==1) or (saco_1==11) or (saco_1==21) or (saco_1==31) or (saco_1==41) or
(saco_1==51) or (saco_1==61) or (saco_1==71) then
PTP P1 CONT Vel= 100 % PDAT3 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

```

**POSICION DEL
SACO SEGÚN
SACO_1**

```

;;;saco 1 capa 2;

```

```

if (saco_1==6) or (saco_1==16) or (saco_1==26) or (saco_1==36) or (saco_1==46) or
(saco_1==56) or (saco_1==66) then

```

```

;;;saco 5 capa 1;

```

```

if (saco_1==5) or (saco_1==15) or (saco_1==25) or (saco_1==35) or (saco_1==45) or
(saco_1==55) or (saco_1==65) or (saco_1==75) then
PTP P5 CONT Vel= 100 % PDAT10 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

```

**INCREMENTO DE
LAS BASES**

```

;;;saco 5 capa 2;

```

```

if (saco_1==10) or (saco_1==20) or (saco_1==30) or (saco_1==40) or (saco_1==50) or
(saco_1==60) or (saco_1==70) then
PTP P10 CONT Vel= 100 % PDAT11 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

```

```

;;;permiso para sacar palet;

```

```

wait sec 0
if (saco_1==sacos_1) then
pulse ($out[10],true,2)
saco_1=0
base_data[3]=base_data[1]
wait sec 0.5
endif

```

**COMPROBACIÓN
PALET TERMINADO**

```

;;;incremento saco;

```

```

saco_1=saco_1+1
ptp_rel{z 110}c_ptp

```

**INCREMENTO DE
LA VARIABLE SACO**

```

;;;punto elevado encima de zona 2

```

```

SYN PULSE 17 'Abrir_Manos' State= TRUE Time= 0.5 sec at START Delay= 0 ms

```

```

PTP P16 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

```

PTP P11 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P24 CONT Vel= 100 % PDAT24 Tool[1] Base[0]
PTP P25 CONT Vel= 100 % PDAT25 Tool[1] Base[0]

```
;verificacion manos y dedos abiertos
continue
loop
continue
if (($in[27]==false) or ($in[28]==false)) then
$timer_stop[1]=false
endif
continue
if $timer[1]>=4000 then
wait sec 0
$timer_stop[1]=true
$timer[1]=0
pulse ($out[12],true,2)
endif
continue
if (($in[27]==true) and ($in[28]==true)) then
$timer_stop[1]=true
$timer[1]=0
exit
endif
endloop
END
```

**VERIFICACION
REPOSO DE LA
MORDAZA**

Para el ejemplo del subprograma **Zona_1 ()** en el que se paletizan 5 sacos por capa, vemos bien diferenciadas las partes de las que se compone. En la parte de cabecera se define mediante la variable constante **alt_incr_1 = 190** el valor a incrementar en la base cada vez que se complete una capa de 5 sacos. Mediante la variable **saco_1** se determina la condición **IF** que contiene el punto con respecto a la **base_data[3]** en la que se dejará el saco, de manera que cuando esta variable valga un múltiplo de 5; es decir , 5,10,15,20,25,30,35..... Se incrementará la base de la siguiente manera:



base_data[3].z=base_data[3].z+alt_incr_1+altura_1

La coordenada Z de la base será la constante de altura declarada al principio del programa + el valor a incrementar establecido desde el XBT. De esta manera cuando vuelva a entrar en el subprograma de zona, posicionará el siguiente saco el valor incrementado por encima.

En la parte final del programa se compara el valor de **saco_1** con el los sacos totales **sacos_1** de manera que si coincide, se finalizará el palet y el KRC no entrará en dicha zona hasta que se genere un palet nuevo, en caso contrario se incrementará el valor de **saco_1** para proseguir con el paletizado.

6.2.2.2 Otras Variables.

Dentro de este apartado se describirán las señales de programa denominadas como bit's o marcas de memoria. Hay disponibles 1024 bit's de entrada que van desde **\$IN[1]...\$IN[1024]** y 1024 bit's de salida que abarcan desde **\$OUT[1] ... \$OUT[1024]**. De todas estas variables se han descritos aquellas que son direccionadas mediante el módulo bechhoff. El resto pueden ser usadas como marcas de memoria teniendo en cuenta que algunas de ellas se encuentran reservadas por el sistema y no podrán ser usadas.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

- **\$OUT[25] -> Restos** . Esta señal será activada desde el programa multitarea y se pondrá a false desde el programa de paletizado tras la notificación al PLC de fin de palet en ambas líneas, y el reinicio de las variables de paletizado.
- **\$OUT[30] -> Simultaneidad de líneas.** Esta señal será usada por el programa para que una vez depositado un saco en una línea, el próximo lo deposite en la otra, de manera que se paletice por igual en ambas líneas. Para ello esta marca se activará en **Zona_1()**, **Zona_11()** y **Zona_111()** y se desactivará en **Zona_2()**, **Zona_22()** y **Zona_222()**. De manera que una vez depositado un saco en una línea la próxima vez lo haga en la otra.

```

;=====ZONA PALETIZADO 1=====
continue
if $out[25]==false then
continue
if (($in[11]==true) and ($in[13]==true) and ($out[30]==false)) then
zona_11 ()
endif
endif

;=====ZONA PALETIZADO 2=====
continue
if $out[25]==false then
continue
if (($in[11]==true) and ($in[14]==true) and ($out[30]==true)) then
zona_22 ()
endif
endif

```

**DENTRO DEL BUCLE
MOSAICO_4 ()**

En el ejemplo observamos la variable **\$OUT[30]** de manera que si en **zona_11()** se activa, la próxima vez dentro del bucle del programa **MOSAICO_4()** sólo podrá entrar en **zona_22()** donde se desactivará para que pueda volver a entrar en **zona_11()** y así sucesivamente.

Hay que tener en cuenta, la posibilidad de que el operario tenga bloqueado mediante selector el no paletizar en una línea; es decir, el PLC no daría permiso de descarga en esa línea de paletizado, de manera que siempre ha de entrar a paletizar en la misma zona. En este sentido desde el programa multitarea, se gestiona esta acción.

```

;=====SIMULTANEIDAD DE LINEA=====

if (($in[13]==true) and ($in[14]==false) and ($out[30]==true)) then
$out[30]=false
endif

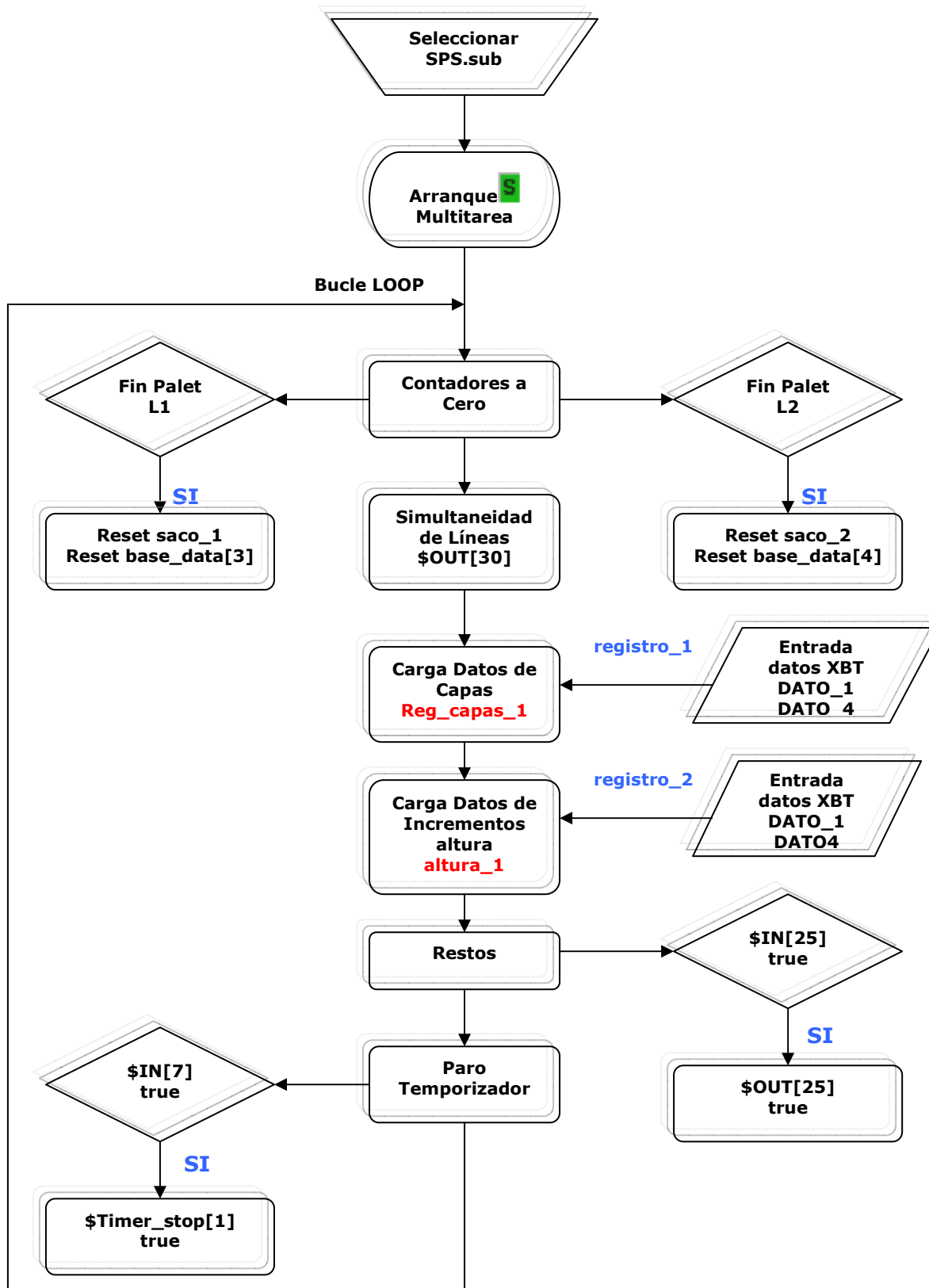
if (($in[13]==false) and ($in[14]==true) and ($out[30]==false)) then
$out[30]=true
endif



```

Si No hay permiso de descarga en la Línea 2, siempre que en **zona_1**, **zona_11** o **zona_11** se ponga la señal **\$OUT[30] = true**; en el submit se volverá a poner al estado false, de manera que se vuelva a entrar a **zona_1**. Lo mismo ocurre si no hay permiso de descarga en línea 1, si **\$OUT[30] = false**, en el submit (programa multitarea) se volvería a poner al estado true para que siempre entre a paletizar en la **zona_2**.

6.2.3 Diagramas de estado software del robot.

6.2.3.1 Interpretador Submit (Multitarea SUB.sps).



	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

El interpretador del submit (**Ver ANEXO SPS.sub**) como hemos venido definiendo a lo largo del proyecto, es un programa multitarea de ejecución cíclica que requiere estar en **RUN** para el funcionamiento del **KRC**, este arranque se encuentra configurado de manera automática, pero es el operario final quién mediante su acción seleccionará el submit y lo arrancará, no necesitando en ningún momento del proceso pararse, ano ser que este provocado por condiciones anómalas de funcionamiento.

El programa se encuentra identificado por una serie de procesos cíclicos, integrados en un bucle infinito donde se repiten por cada período de reloj. Tal como se ha ido detallando estos procesos se ejecutan de manera cíclica. En el primero de ellos se evalúa la señal de fin de palet de cada una de las líneas, de manera que en caso de finalizado el palet, el programa multitarea reiniciará la variable de posición de saco y la coordenada Z de la base_data.



En el segundo de los procesos, se evalúa la señal de simultaneidad de líneas, de manera que se asegura que si una de la líneas no tiene permiso de descarga por medio del PLC la señal **\$OUT[30]** permanecerá activada o desactivada consintiendo la descarga sobre la otra línea de paletizado. Este proceso solamente se evaluará si el PLC, quita permiso de descarga a alguna de las líneas de paletizado.

En los procesos de carga de datos, se captura los valores **registro_1** y **registro_2** por medio de los bit's **DATO_1 to DATO_4** introducidos por pantalla en cuanto al número de alturas a paletizar y los incrementos. En el submit se generan los valores **reg_capas_1**, que determina el número de capas a paletizar cuando el programa se encuentre en **home()**, y el incremento de alturas **altura_1** que se suma al incremento de Z constante por cada capa finalizada.

Restos simplemente captura la pulsación del armario eléctrico para tal fin, de manera que el robot finalizado los movimientos en ejecución del programa de paletizado actual, activará las señales de fin de palet en ambas líneas, y volverá posteriormente a **false** el estado de dicha señal.

El último de los procesos a llevar a cabo, es el paro de la temporización para activar el defecto de mordaza no en reposo; esto es, por que si el operario para la instalación mediante el pulsador de paro, se deshabilita la señal de habilitación de movimientos **MOVE_ENABLE->\$IN[7]**, de esta manera si detenemos la cuenta del temporizador este no saltará la alarma con la instalación parada.

Finalizados todos los procesos, se ejecutaran todos y cada uno de ellos desde el principio, siendo este un bucle infinito mientras ningún ser humano detenga el interpretador.

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

Básicamente queda representada la estructura de programación del programa **Cell.src**, que al igual que el **SUB.sps**, tendrá que ser seleccionado y arrancado por primera vez mediante intervención del operador.

En el momento que el programa es seleccionado, habrá que realizar el arranque en RUN del programa mediante la comunicación con el PLC. A través de la Interface del Automático externo procederemos al rearme de los accionamientos del robot y la confirmación de mensajes, tal como se desarrolló en los procedimientos anteriores.

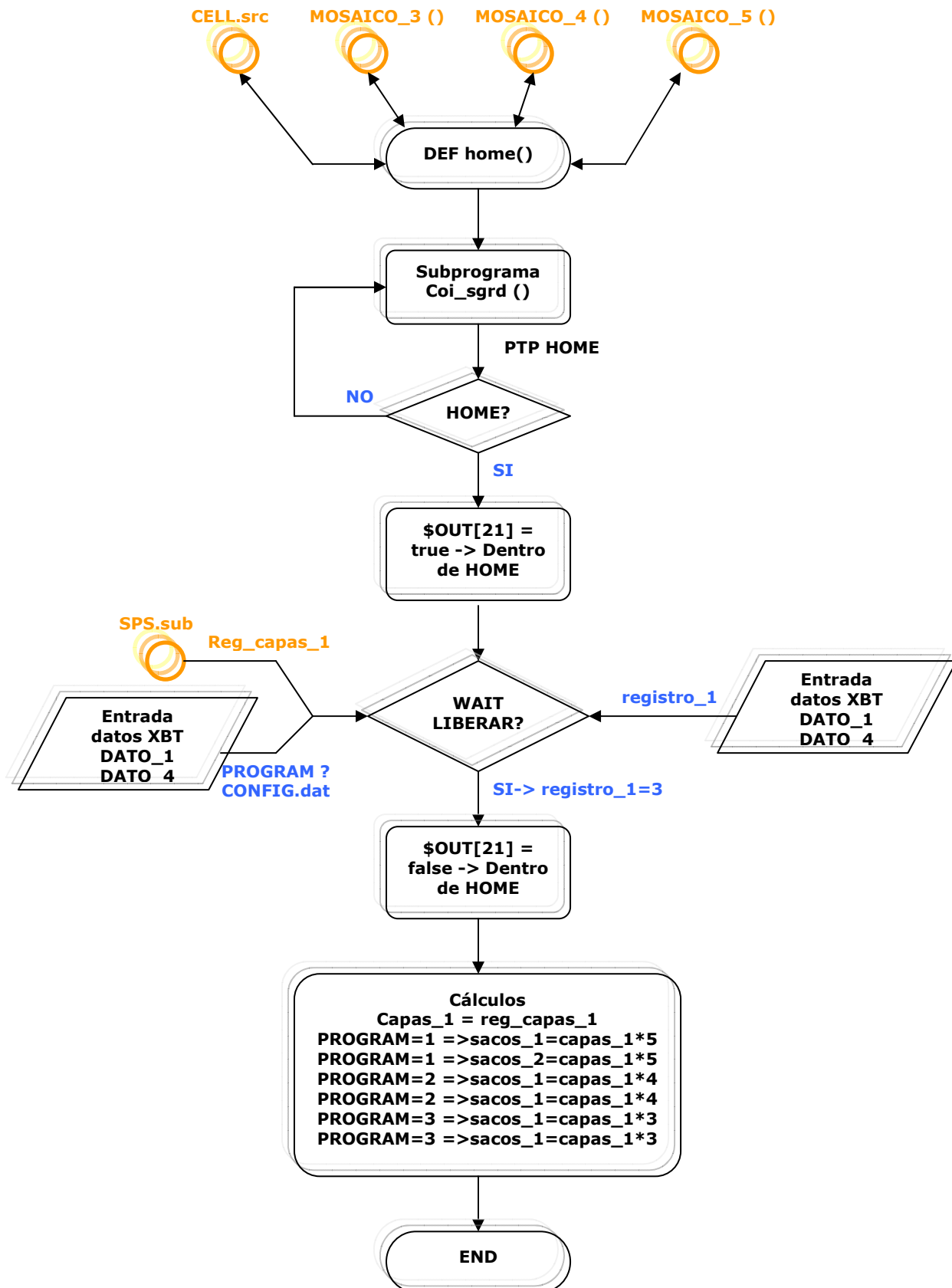
Una vez que el programa esta en RUN, se inicializa la cabecera con la declaración de los programas a usar por **CELL.src**, seguidamente se hace es una coincidencia de trayectorias COI, y la selección del programa de paletizado que queremos usar. Para ello se llama al proceso **home ()**.



Dentro de Home se llama al subproceso **coi_sgr ()** que sitúa al robot en un punto seguro en espera de la recepción de los datos de paletización. Finalizado el proceso **coi_sgr ()** se activará la marca correspondiente indicando la posición de HOME al PLC. En este momento podremos transmitir desde el terminal XBT el valor de programa que será transmitido por el PLC mediante las variables PGNO_BIT y convertido por la variable de tipo signal PROGRAM en el **CONFIG.dat**. Después de transmitido el programa a seleccionar y las alturas totales que captura el proceso **home ()** a través del programa multitarea, esperaremos en dicha posición a ser liberado desde el terminal.

Cuando el robot es liberado se calculan las variables enteras **sacos_1** y **sacos_2** que contienen la cantidad de sacos totales que queremos paletizar en las dos líneas por igual.

Seguidamente se entra en el bucle de ejecución infinito, en donde se activa el programa de Interface **P00.src** que contiene las funciones necesarias para realizar la comunicación con el PLC, en el intercambio de datos y a través de las líneas destinadas para ello. De esta manera podremos realizar una nueva petición y envío de datos en cada cambio de programa. Dentro de este mismo bucle se ejecuta la función **CASE (Switch)**, en donde en función de la variables **PGNO_BIT_1**, **PGNO_BIT_2** y **PGNO_BIT_3** se ejecutará el proceso **Mosaico_5 ()**, **Mosaico_4 ()** o **Mosaico_3 ()**. Estos procesos a su vez contienen un bucle de ejecución infinita con la posibilidad de salir de ellos en caso de petición de un nuevo programa como veremos más adelante.

6.2.3.3 Home.src.



	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

El subprograma **home ()** es un proceso desde el que tienen acceso los programas principales para la petición de cambio del número de alturas o del programa a ejecutar. También tiene acceso **cell.src** en el proceso de arranque.

Lo primero que se ejecuta es el subproceso **coi_sgrd ()** para la coincidencia de un punto seguro de trayectoria. En el momento que el robot adquiere la posición de Home, activará la señal **\$OUT[21]** para notificárselo al PLC, y pasará al modo de espera hasta el envío de la información necesaria y la posterior liberación.

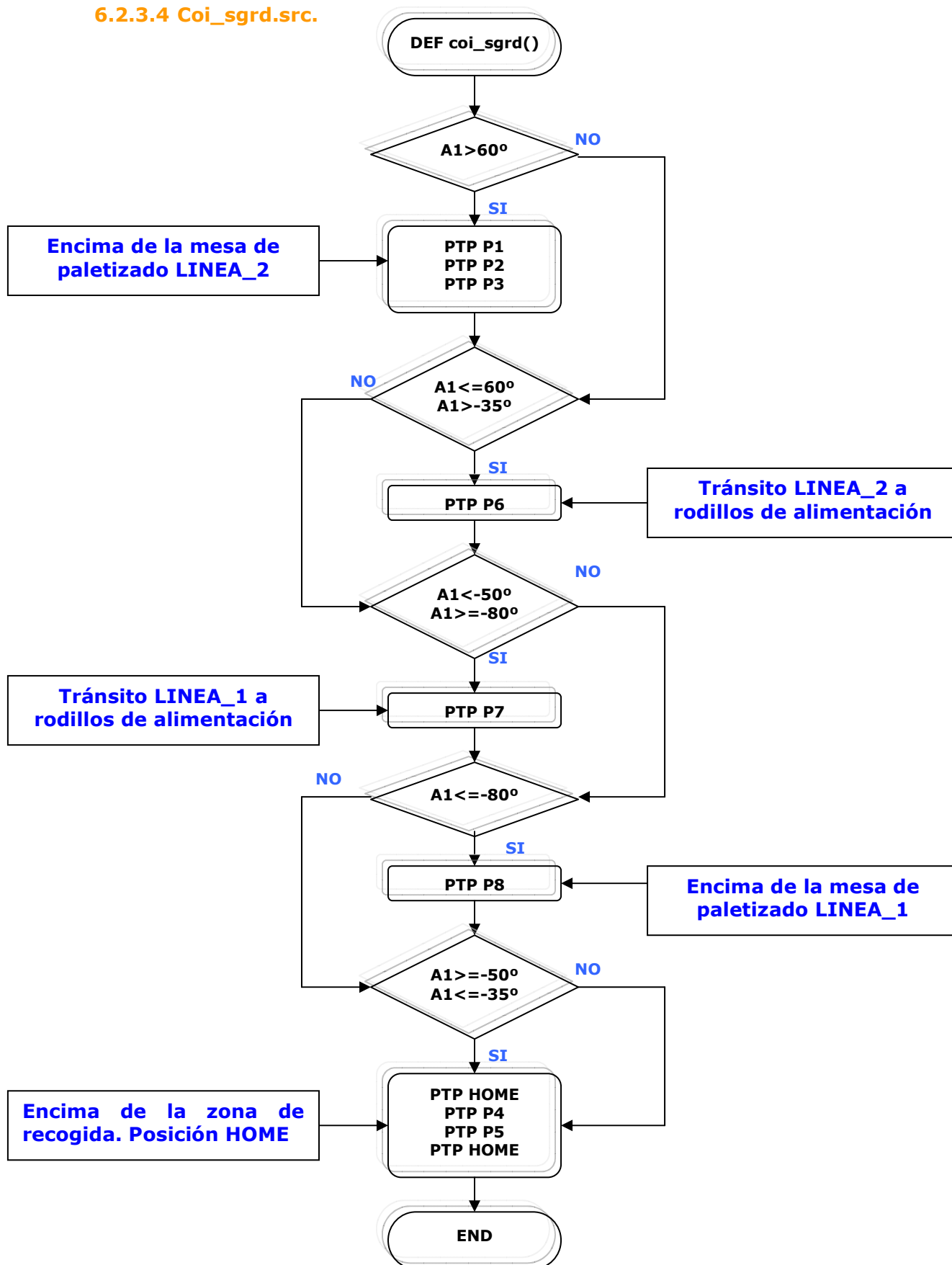
En el modo de espera, enviaremos el número de alturas desde el terminal XBT, en caso de no enviar ninguna altura, en la variable **reg_capas_1** capturada en el programa multitarea, estará guardado el último valor enviado. Antes de liberar, también tendremos que enviar desde el XBT el programa de mosaico que queramos realizar. Dicho programa quedará guardado en la variable **PROGRAM** del **Config.dat** que podremos usar en cualquier momento.


Una vez enviado el tipo de mosaico y el número de alturas adecuado, liberaremos al robot desde la función provista en el XBT para tal fin. La condición de liberación pone la variable **registro_1 = 3**. Además, al ser una orden de liberación, las entradas del robot adquieren los valores necesarios para tal fin, siendo la condición de liberación la siguiente línea de código:

wait for ((\$in[17]==false) and (\$in[18]==false) and (\$in[19]==true) and (\$in[20]==false) and (registro_1==3)).

Una vez cumplida la condición de liberación pondremos a false la señal **\$OUT[21]** puesto que el valor de programa ya ha sido transmitido, y pasaremos a realizar los cálculos totales de las variables **sacos_1** y **sacos_2** que determinarán el número máximo de sacos a paletizar en el palet antes de su finalización. Estos valores quedarán actualizados en el fichero **Config.dat**.

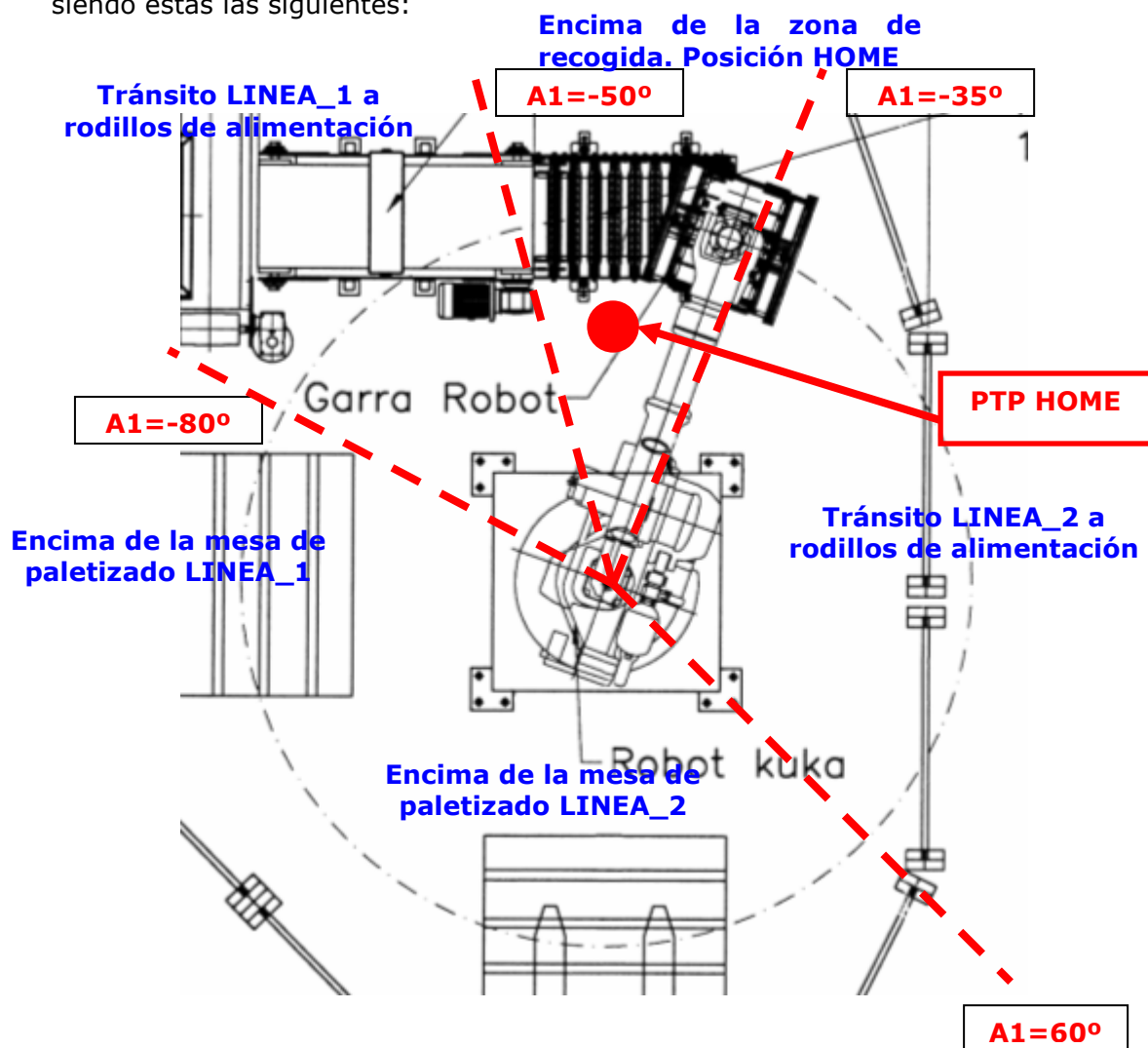
6.2.3.4 Coi_sgrd.src.



ROBITEK.System Dpto.Robótica	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

El subprograma **coi_sgrd.src** siempre será llamado desde el proceso **home()**, aunque también será llamado en el arranque de los programas de paletizado principales **MOSAICO_3()**, **MOSAICO_4()** y **MOSAICO_5()** por seguridad.

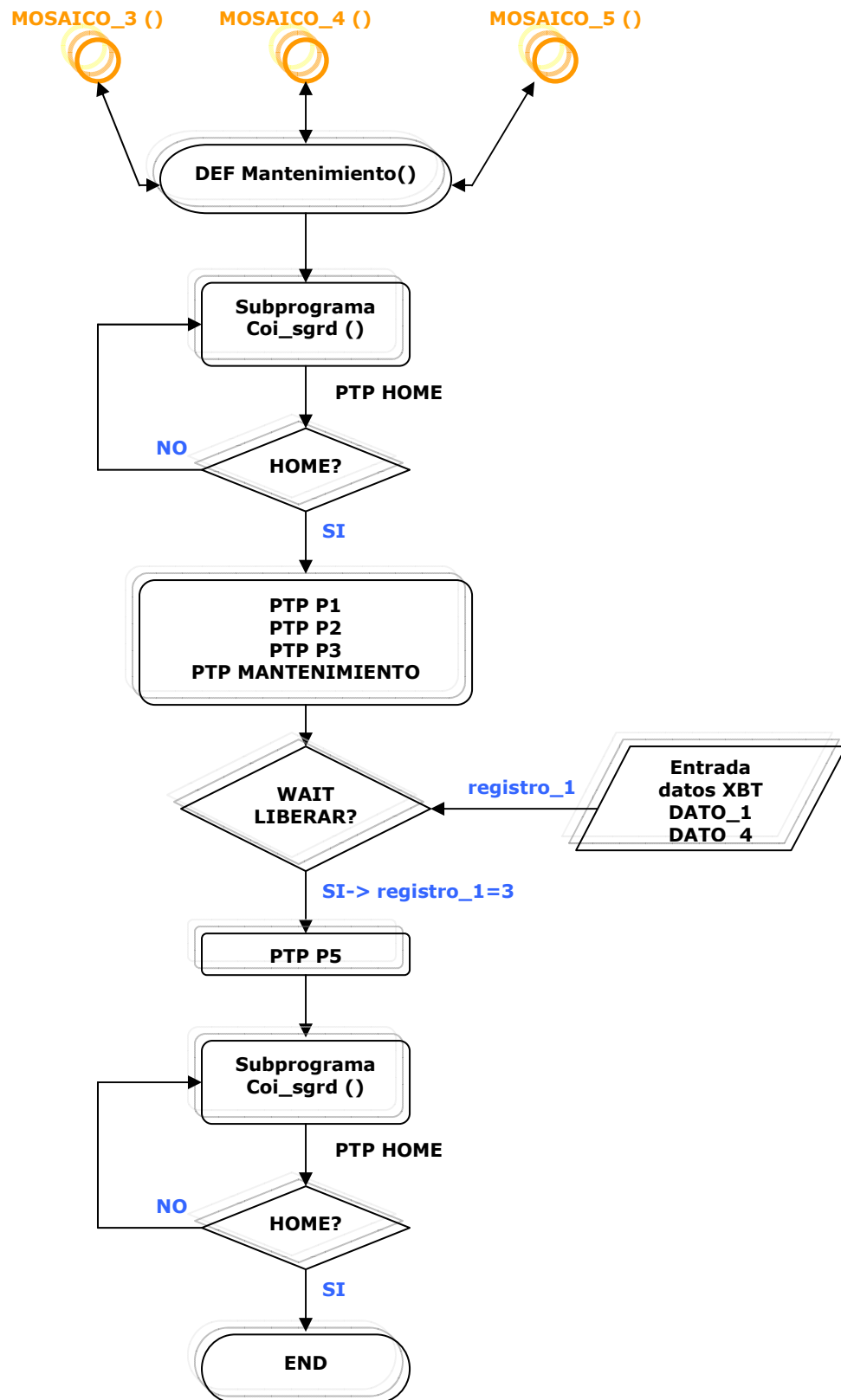
Lo que se va realizando es una comprobación del eje principal **a1** del robot, de manera que en función de su posición se le va llevando a una serie de puntos hasta alcanzar el punto HOME de destino. Para ello se recorren todas las zonas siendo estas las siguientes:





Para el caso de estar encima del palet de la línea 2, el ángulo del eje principal es **a1 > 60°** en este caso se genera la trayectoria **PTP P1, P2 y P3** para situar al robot en una zona libre de obstáculos. Si el robot se encuentra entre los ángulos **a1 ≤ 60° y a1 > -35°** corresponde a una zona de tránsito a los rodillos de alimentación, por lo tanto se ha de situar al robot en el punto **PTP P6**. De esta manera en función de los grados de desvío del robot se determina una posición auxiliar hasta alcanzar el punto exacto de reposo **PTP HOME**. Los puntos **PTP P4** y **PTP P5** son exactamente el punto home situado por encima y por debajo del mismo, de esta manera cada vez que el robot haga este gesto, le indicará visualmente al operador que se ha alcanzado la posición solicitada.

Seguidamente se continuará con el proceso desde el cual fue llamado **coi_sgrd ()**.

6.2.3.5 Mantenimiento. src



	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

El subprograma de mantenimiento es un sencillo grafo al que tienen acceso los programas principales de paletizado. El robot se sitúa en un punto de acceso donde el operador de mantenimiento tiene la posibilidad de verificar el estado de los componentes de la garra.

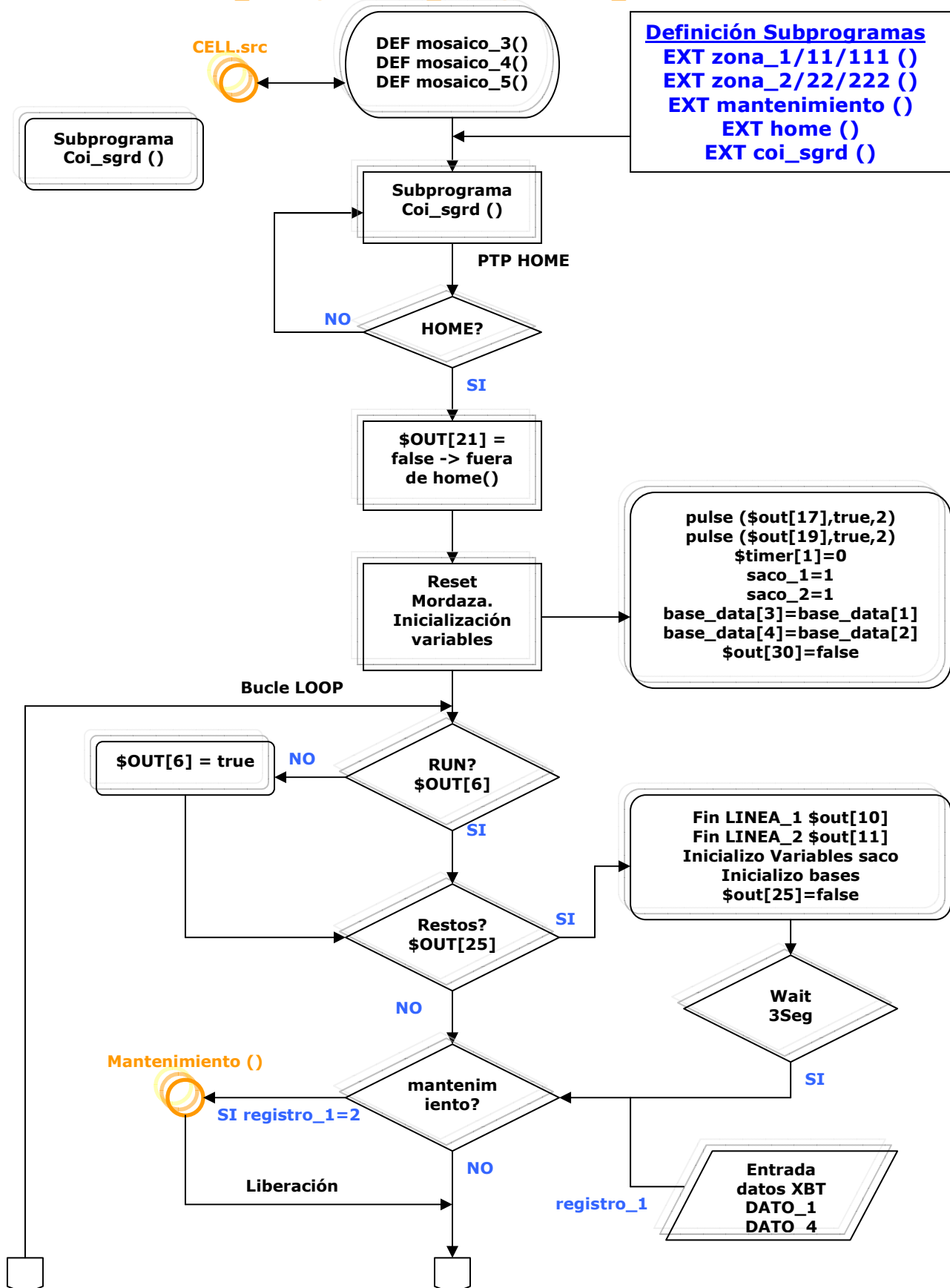
Lo primero que se ejecuta por seguridad es el subproceso **coi_sgrd ()** para colocar al robot en un punto seguro dependiendo de la situación del eje principal. Una vez situado el robot en su posición de HOME, el robot se moverá a velocidad reducida hasta un punto de acceso dentro de la periferia.

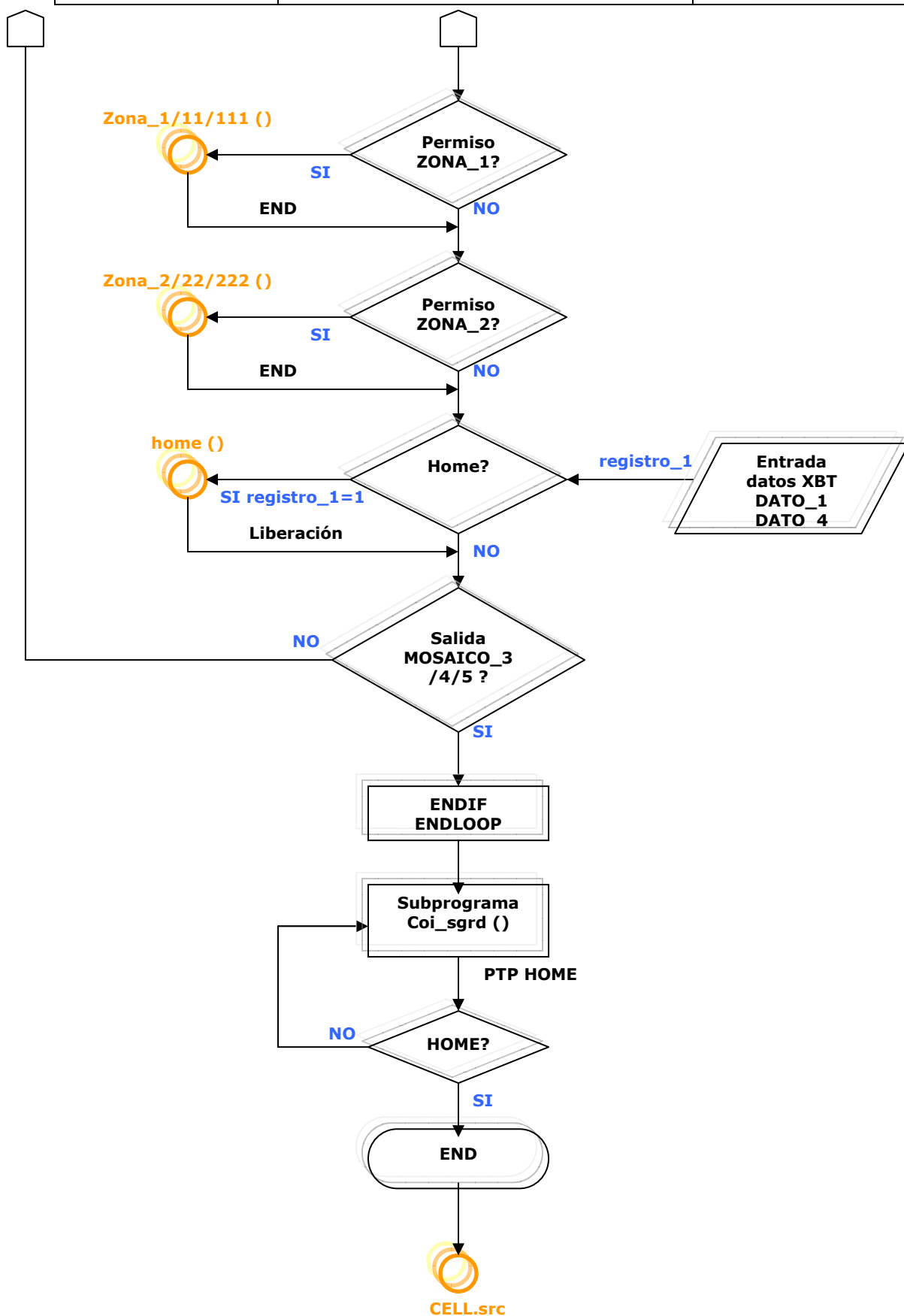
El robot se detendrá y permanecerá en dicha posición hasta que no liberemos desde el terminal XBT. Durante este tiempo el personal cualificado podrá entrar al recinto, saltando las seguridades de la máquina para poder trabajar de forma segura.



Una vez finalizados los trabajos de verificación, habrá que rearmar nuevamente las seguridades de la instalación, y una vez puesta en marcha procederemos a liberar desde el XBT.

Después de liberado, el robot se situará nuevamente en su posición HOME, y continuará con el proceso desde el que fue llamado.

6.2.3.6 Mosaico_3.src / Mosaico_4.src / Mosaico_5.src





	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

Los programas principales, son exactamente iguales cambiando los subprocesos de zona. Por lo tanto tendremos:

- **MOSAICO_3()** {
 - Zona_111 ()**-> Paletizado en la línea 1.
 - Zona_222 ()**-> Paletizado en la línea 2.

- **MOSAICO_4()** {
 - Zona_11 ()**-> Paletizado en la línea 1.
 - Zona_22 ()**-> Paletizado en la línea 2.

- **MOSAICO_5()** {
 - Zona_1 ()**-> Paletizado en la línea 1.
 - Zona_2 ()**-> Paletizado en la línea 2.

Una vez que el proceso es llamado desde el cell.src, lo primero que se ejecuta es el punto seguro de trayectoria por medio de **coi_sgrd ()** donde si el robot ya se encuentra en el punto HOME, se le notificará al operario mediante los movimientos arriba y abajo.

Seguidamente, y antes de iniciar el bucle de ejecución cíclico, se hace un reset de las variables de programa y de la mordaza esto es:

- **\$out[21]=false**-> Al no estar dentro del subproceso HOME

Para asegurar el estado de reposo de la mordaza.



- **pulse (\$out[17],true,2)** -> Apertura de las manos de la garra.
- **pulse (\$out[19],true,2)** -> Apertura de los dedos de la garra.

- **\$timer[1]=0**-> Puesta a cero del temporizador para el defecto de la mordaza.
- **saco_1=1**-> Inicialización de la variable de posición para la línea_1.
- **saco_2=1**-> Inicialización de la variable de posición para la línea_2.
- **base_data[3]=base_data[1]**-> Inicialización de la base en la línea_1.
- **base_data[4]=base_data[2]**-> Inicialización de la base en la línea_2.
- **\$out[30]=false**-> Reset de la marca de simultaneidad de líneas.

Después de reinicializadas las señales que entraran en juego en el proceso de paletizado, se entra en un bucle de ejecución infinita, en el que permaneceremos hasta que se cumplan las condiciones necesarias para el cambio de programa de paletizado. Dentro de este bucle se evalúan por orden descendente los siguientes procesos:

1. Comprobación del estado RUN de programa.

Con el programa de paletizado en marcha, el KRC pone a true la variable de sistema automático externo **APPL_RUN -> \$OUT[6]** para notificárselo al PLC, y así seguir con lo establecido en el programa del PLC. En algunas circunstancias es posible que el propio sistema no mantenga esta señal a true por posibles conflictos, reinicio inesperados o que el propio operador la haya forzado a false. Para asegurarnos del correcto funcionamiento de esta señal se evalúa su estado **true** o **false**, y en caso de no encontrarse a **true** una vez habilitado el

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

movimiento; esto es **MOVE_ENABLE->\$IN[7]=true** y con el selector en modo automático externo, se forzará a su estado **true** por programa. En caso de que su estado sea el correcto se continúa con el siguiente proceso.

2. Comprobación de la señal de RESTOS.

En cada ciclo dentro del bucle y antes de comprobar los procesos de zona_1 y zona_2, se comprueba si la señal de restos ha sido accionada mediante el valor **\$OUT[25]** que escribe el programa multitarea submit.

Si esta señal ha sido accionada entraremos en la condición IF, donde se finaliza palet en ambas líneas y se resetean las variables de paletizado y bases de referencia de los palet's. Al cabo de un tiempo de espera establecido en tres segundos. El PLC quitará permiso de descarga en ambas líneas hasta que no se pongan unos palet's nuevos, mientras tanto la línea de programa pondrá a false la señal **\$OUT[25]** dejándola libre para una nueva evaluación de la señal de restos en caso de producirse y continuará con su ejecución cíclica dentro del bucle.

Después terminada la rama condicional IF de los restos, o si estos no se han cumplido, se continúa con el siguiente proceso.

3. Comprobación del estado ir a Mantenimiento.



La condición para entrar en el subproceso de mantenimiento, depende única y exclusivamente de la orden generada desde el terminal XBT. Esta orden puede ser ejecutada en cualquier momento del programa. Lo único a tener en cuenta es que si generamos esta orden desde el XBT, en el momento que el robot está ejecutando movimientos de dejada de saco dentro de zona_1() o de zona_2 (), es posible que cuando el robot salga de la zona y vuelva a ver la condición IF de entrada a mantenimiento, el PLC; que sólo mantiene un tiempo las señales **\$IN[17], \$IN[18], \$IN[19]** y **\$IN[20]**, ya habrá puesto estas señales a false por lo que no entraríamos dentro de la condición. Lo mejor para ejecutar el proceso de mantenimiento, es que la línea de programa del KRC se mantenga dentro del bucle principal, para lo que podemos cancelar los permisos de entrada a las zonas mediante los selectores, o evitamos la entrada de sacos en los rodillos de alimentación para que no haya permiso de carga, de este modo el robot se mantendrá parado esperando alguna orden y en cuanto ejecutemos la orden desde el terminal se cumplirán las señales poniendo la variable **registro_1** que adquirirá el valor "2" entrando dentro del proceso de mantenimiento.

Liberado el robot de su estado de mantenimiento desde el XBT, o si no se cumple la condición de entrada en mantenimiento, el programa continúa con la siguiente evaluación.

4. Entrada a zona de paletizado en LINEA-1.

La condición para entrar en el subproceso **Zona_1 ()**, **Zona_11()** o **Zona_111()** dependiendo del programa de paletizado que se este ejecutando. Es la siguiente:

- **\$out[25]==false->** La señal de restos no debe estar activada por seguridad.
- **\$in[11]==true->** El PLC debe dar Permiso de carga.
- **\$in[13]==true->** El PLC debe dar permiso de descarga en la línea 1.
- **\$out[30]==false->** La señal simultaneidad de líneas ha de estar desactivada.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

Si esto se cumple, el robot entrará en la zona de paletizado. Cada vez que se entra en la zona de paletizado, se recoge un saco de los rodillos de entrada y se deposita en el palet correspondiente incrementando la variable saco_1, de manera que la próxima vez que entre dentro de la zona el saco lo dejará en la siguiente posición en el palet.

Finalizada las acciones de recogida del saco y dejada en el palet, o en el caso de que no se cumplieran las condiciones anteriores, se procede a continuar con el siguiente proceso en ejecución dentro del bucle.

5. Entrada a zona de paletizado en LINEA-1.

La condición para entrar en el subproceso **Zona_2 ()**, **Zona_22()** o **Zona_222()** dependiendo del programa de paletizado que se este ejecutando. Es la siguiente:

- **\$out[25]==false->** La señal de restos no debe estar activada por seguridad.
- **\$in[11]==true->** El PLC debe dar Permiso de carga.
- **\$in[14]==true->** El PLC debe dar permiso de descarga en la línea 2.
- **\$out[30]==true->** La señal simultaneidad de líneas ha de estar activada.

Si esto se cumple, el robot entrará en la zona de paletizado. Al igual que en el caso anterior se recoge un saco de los rodillos de entrada y se deposita en el palet correspondiente incrementando la variable saco_2, de manera que la próxima vez que entre dentro de la zona el saco lo dejará en la siguiente posición en el palet.



Finalizada las acciones de recogida del saco y dejada en el palet, o en el caso de que no se cumplieran las condiciones anteriores, se procede a continuar con el siguiente proceso en ejecución dentro del bucle.

6. Entrada a HOME para solicitud de cambio de alturas o de Programa.

La entrada a Home, la realizaremos cada vez que queramos cambiar el mosaico de paletizado, o sin cambiar el programa de paletizado pero modificando la altura máxima a paletizar. Será por tanto como venimos comentando, que las líneas de paletizado estén vacías, esto es necesario bien para que se empiece el nuevo mosaico desde la primera altura, o bien para no descuadrar la posición de los sacos y cambiar las altura del mismo programa de paletizado. La condición será por lo tanto la siguiente:

- | | | |
|--|---|---|
| <ul style="list-style-type: none"> -\$in[17]==false -\$in[18]==false -\$in[19]==true -\$in[20]==false -registro_1==1 | } | Orden de ir a HOME desde el terminal XBT. |
| <ul style="list-style-type: none"> -saco_1<2 -saco_2<2 | | |

Las variables de posición de saco saco_1<2 y saco_2<2 indica que su valor solamente puede ser cero o uno, de manera que esto indica que el palet está vacío. Bien por una finalización de palet, o bien por el accionamiento de restos, de manera

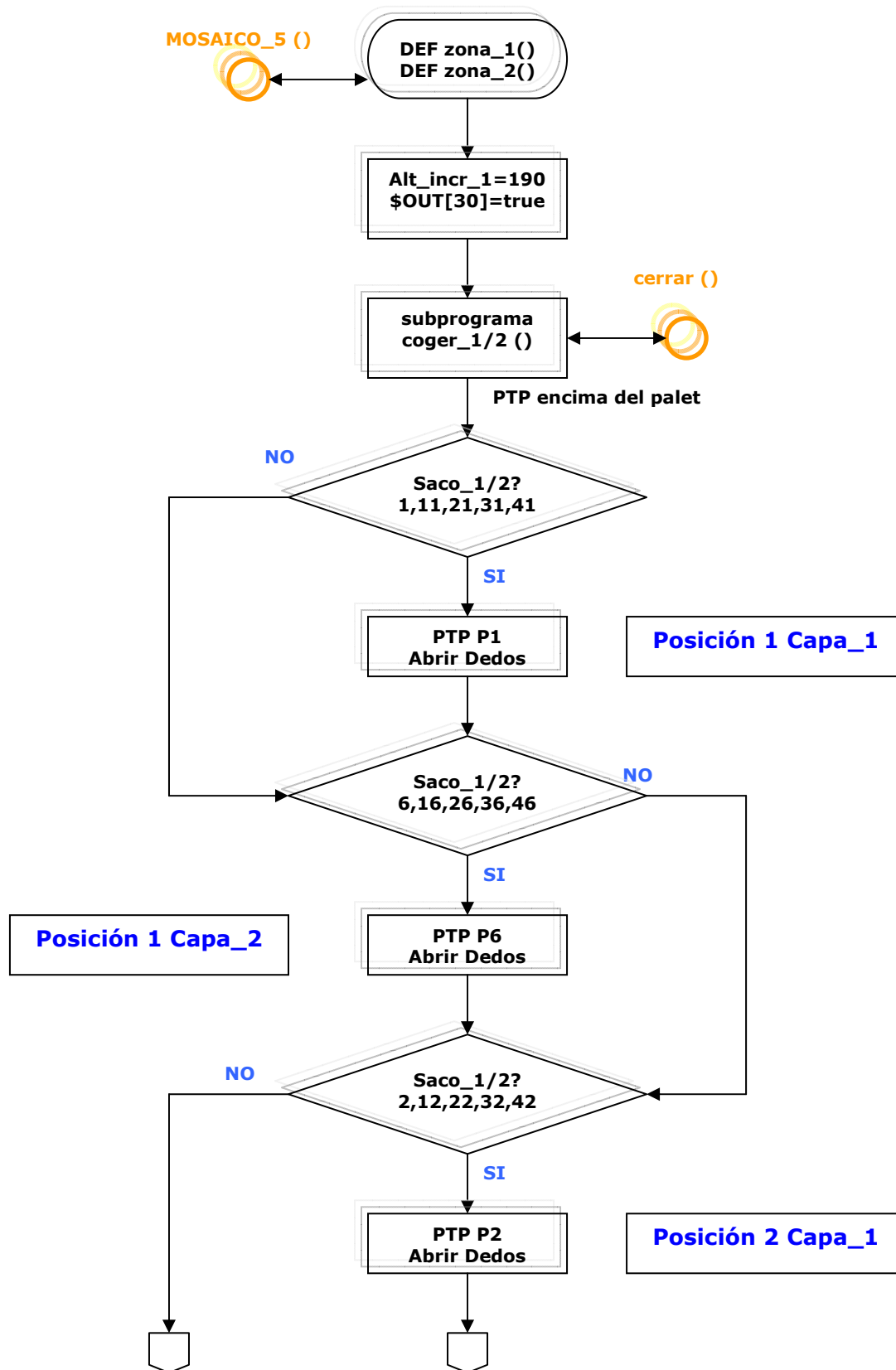
	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

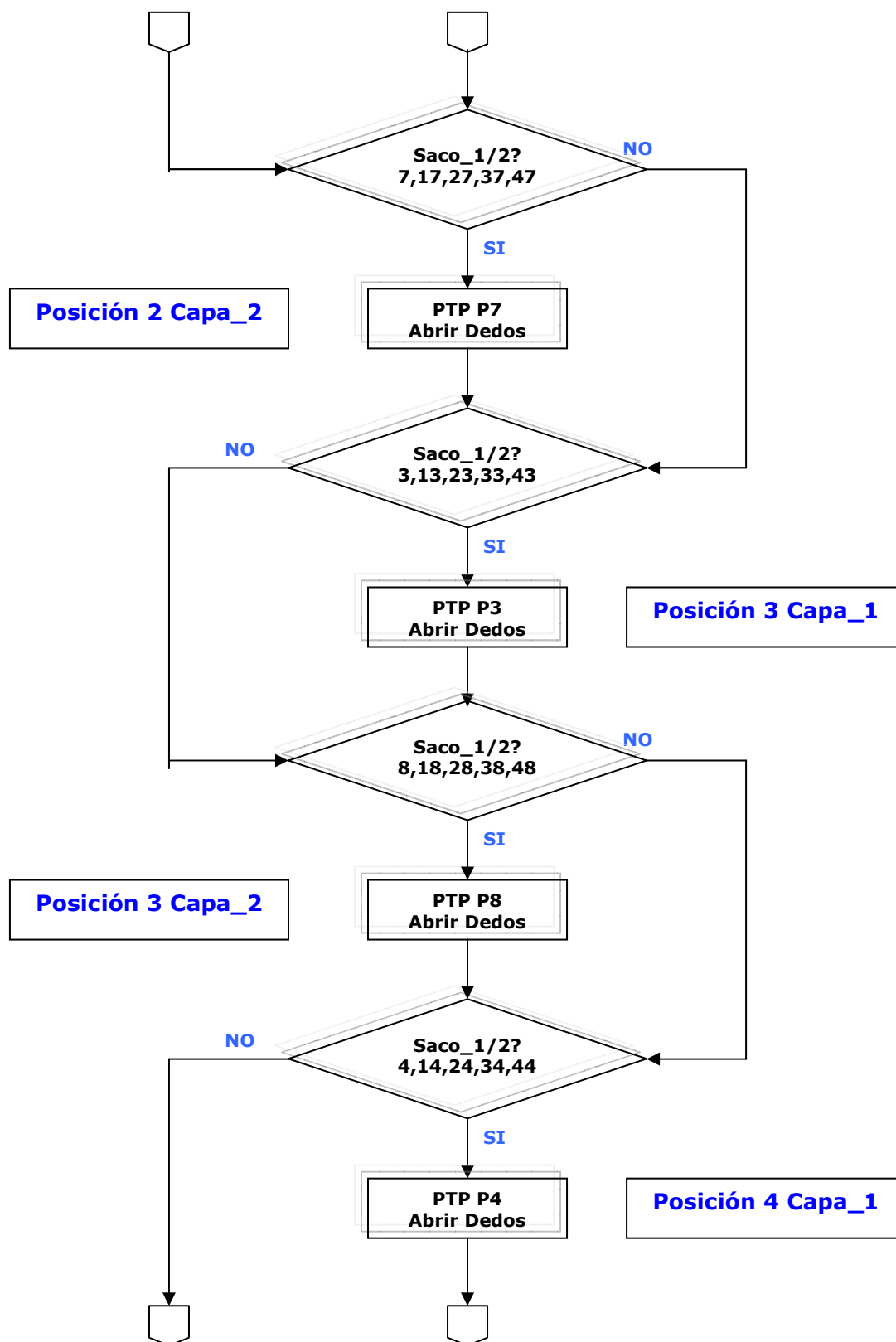
que siempre será preciso sacarlo y generar un nuevo palet mediante los procedimientos establecidos.

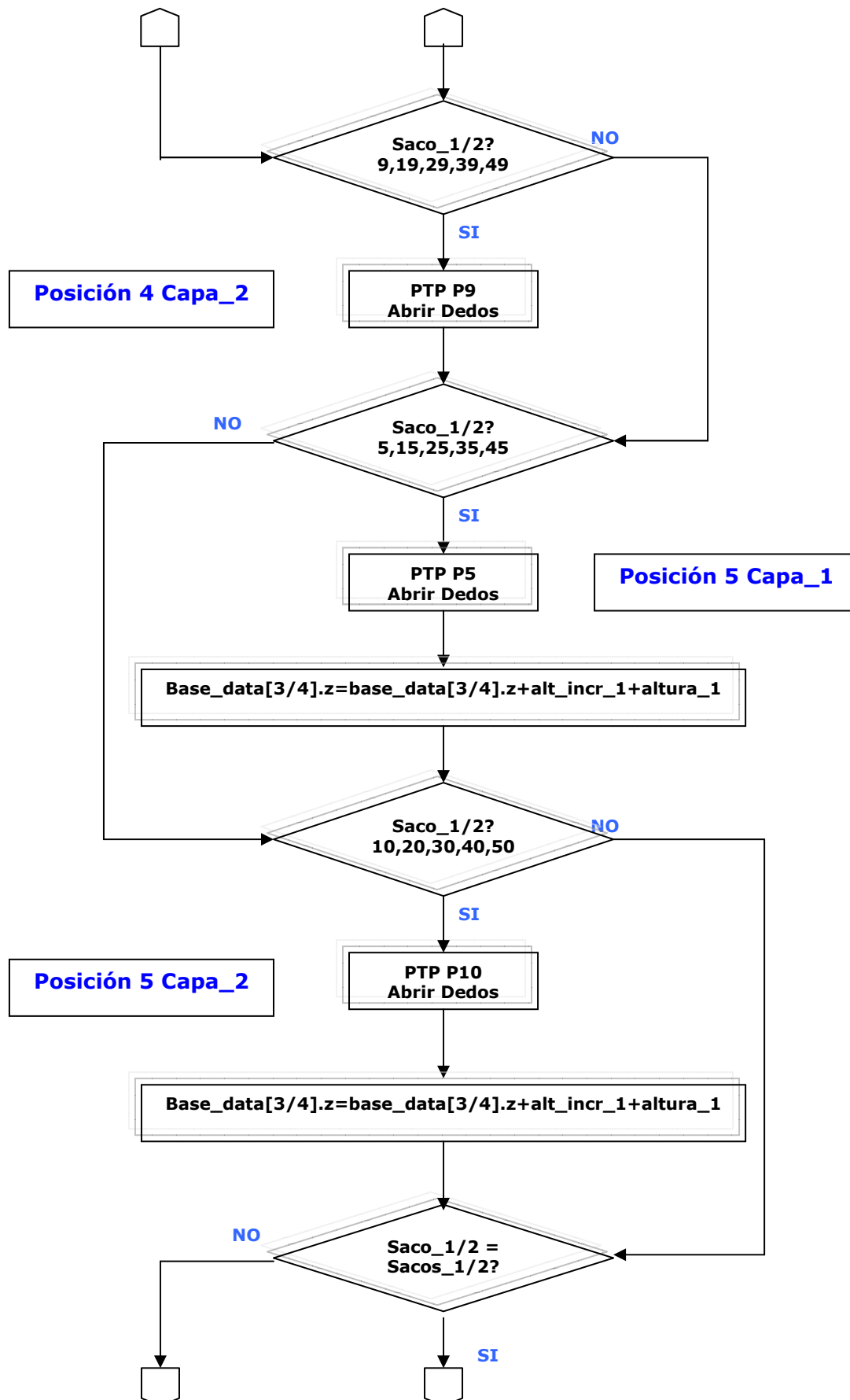
La entrada a HOME se realiza como último proceso del bucle de ejecución, de manera que si al salir de HOME hemos cambiado la variable **PROGRAM**, no coincidiendo con el programa en ejecución actual, se saldrá del bucle mediante la función **EXIT**, no volviendo a paletizar en las zonas correspondientes al mosaico actual. A parte de tener en cuenta el cambio de programa en la salida condicional del bucle, se precisará que no entren sacos en los rodillos de alimentación; es decir, no tener permiso de carga.

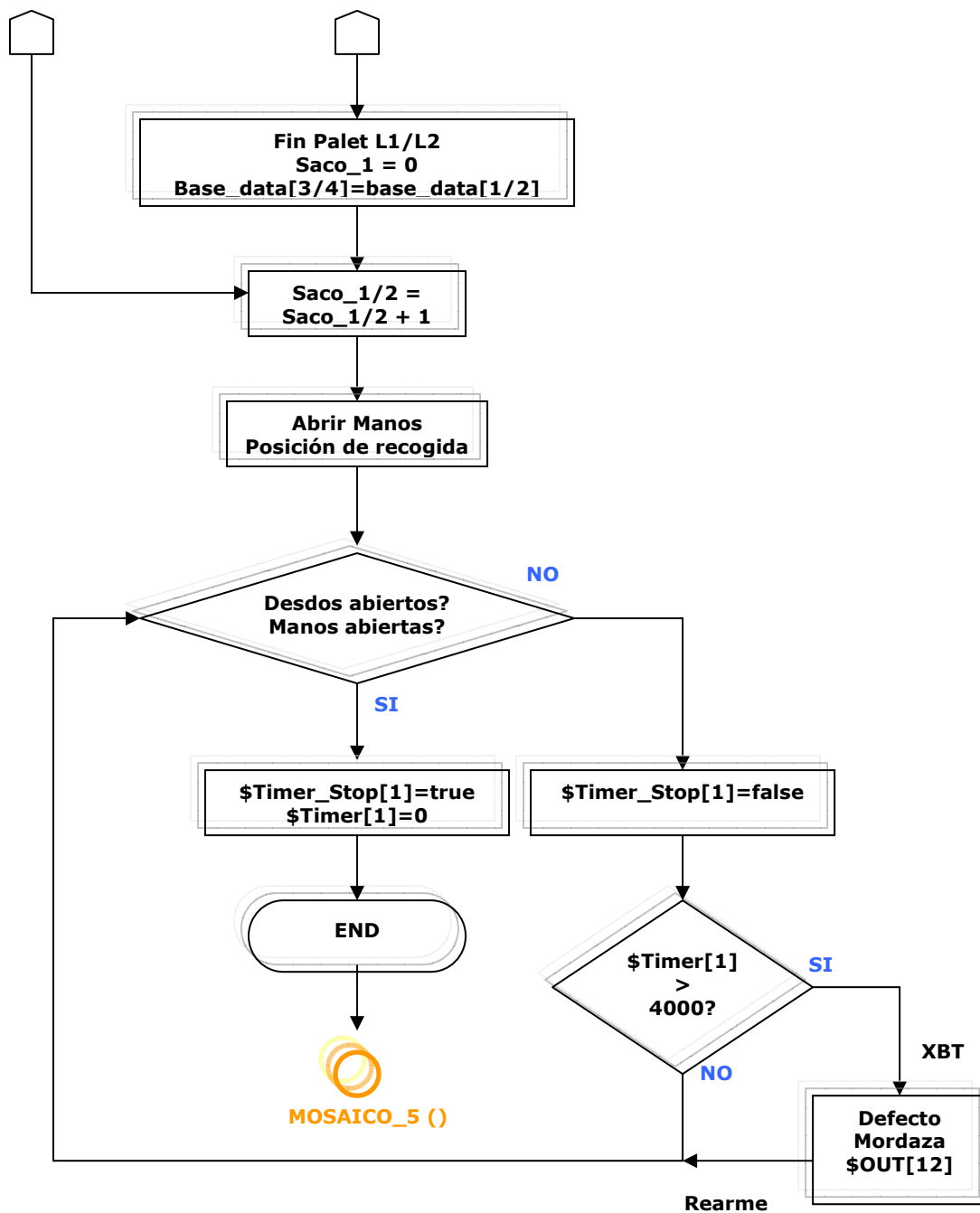
Una finalizado el bucle, se vuelve a ejecutar una coincidencia de trayectoria, de manera que cuando el robot ejecute los movimientos arriba y abajo del Home, indicará al operario que este ha salido del programa actual y a entrado en el nuevo. En este momento se admitirá nuevamente la entrada de palet's pero no tendremos permiso de descarga hasta que el operario quite los palet's finalizados con el mosaico anterior.



6.2.3.7 Zona_1.src / Zona_2.src









	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

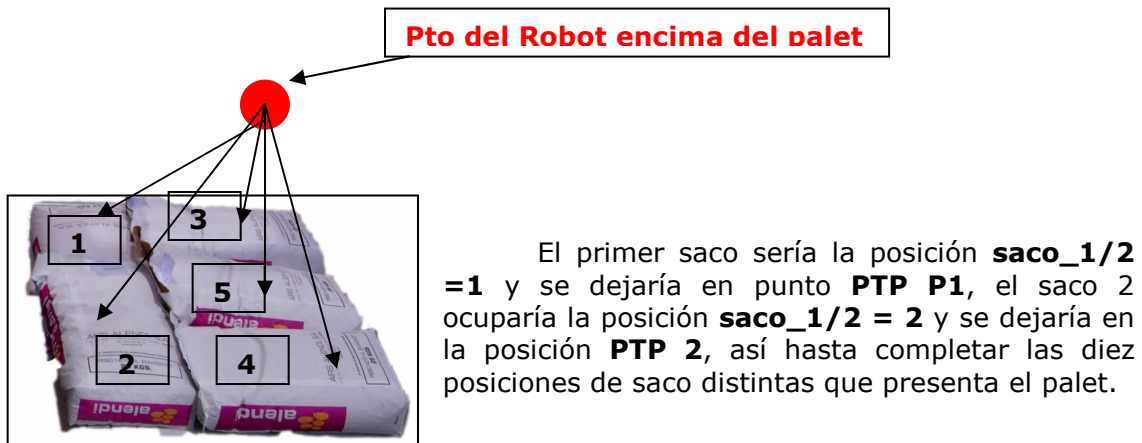
Zona_1 () es el programa de paletizado con mosaico de cinco sacos en la línea 1 y **Zona_2 ()** el de la línea 2 que es exactamente idéntico cambiando la variable de posición de saco, fin de palet, etc. Pero funcionalmente son idénticos, por lo que se explican ambos diagramas en conjunto. Centrándonos en **zona_1/2()** que será llamado desde el bucle infinito del programa principal **MOSAICO_5**.

- **Zona_1/2 () => Zona_1() o Zona_2 ()**.
- **Saco_1/2 => Saco_1 o Saco_2**.
- **Sacos_1/2 => Sacos_1 o Sacos_2**.
- **base_data[3/4] => base_data[3] o base_data[4]**.
- **coger_1/2 () => coger_1 () o coger_2 ()**.



Lo primero que se hace en estos procesos, es asignar un valor a la variable **alt_incr_1** del archivo config.dat, para el caso que nos ocupa **alt_incr_1=190** en milímetros será la altura base a incrementar en la coordenada Z de la base relacionada al palet; **base_data[3]** para la línea 1 y **base_data[4]** para la línea 2, por cada altura completada.

Seguidamente se ejecuta el subproceso **coger_1 () o coger_2 ()**. Mediante la ejecución de este subprograma, el robot ejecuta las trayectorias que van desde la mesa de rodillos hasta un punto situado encima del palet de la línea 1, y a partir del cual se posiciona el saco en el palet.

Como hemos visto en apartados anteriores, para el mosaico de 5 sacos por altura, existen 2 capas diferentes; esto significa, que tendremos 10 posiciones diferentes de sacos a partir de las cuales se irán repitiendo con el incremento en Z correspondiente. Si vemos un ejemplo para la capa 1 del mosaico de 5 sacos, cada saco tiene una posición que sería la siguiente:



Una vez posicionados los diez primeros sacos, el saco número once que ocupará la posición **saco_1/2 = 11** corresponderá con el saco nº 1 de la altura 3, y por lo tanto su punto de dejada será el mismo que el punto **PTP P1** de la altura uno. Cada cinco sacos tendremos que incrementar el valor constante **alt_incr_1**, esto será como veremos, en los sacos múltiplos de 5; es decir **saco_1/2 = 5, 10, 15, 20, 25** y así hasta el final del palet. De esta manera vemos que los procesos de zonas se componen de condiciones IF dependientes de la variable de posición **saco_1** o **saco_2** la cual se irá incrementando por cada saco dejado en el palet.

	Proyecto	
	Documento	

PROYECTO FIN DE CARRERA

Instalación de Robot Paletizado.

Una vez que el robot esta posicionado encima del palet con el saco entre sus garras, se evalúa el valor de la variable **saco_1/2**, de manera que solamente se ejecutará la condición **IF** correspondiente a este valor. Para posicionar un saco el robot ejecutará el punto **PTP (referenciado a la base_data[3] en la línea 1 o base_data[4] en la línea 2)** para posicionarse justo encima de donde debe soltarse el saco, seguidamente antes de abrir los dedos del robot se hará un movimiento relativo en Z de manera lineal a la posición PTP anterior.

Soltado el saco en la posición correspondiente, esperaremos el tiempo que asegura la retirada del robot sin peligro de llevarse el saco que se encuentra en caída. Si el saco depositado es un múltiplo de 5, se incrementará la base del palet, con la constante de altura y el valor configurado a través del XBT.

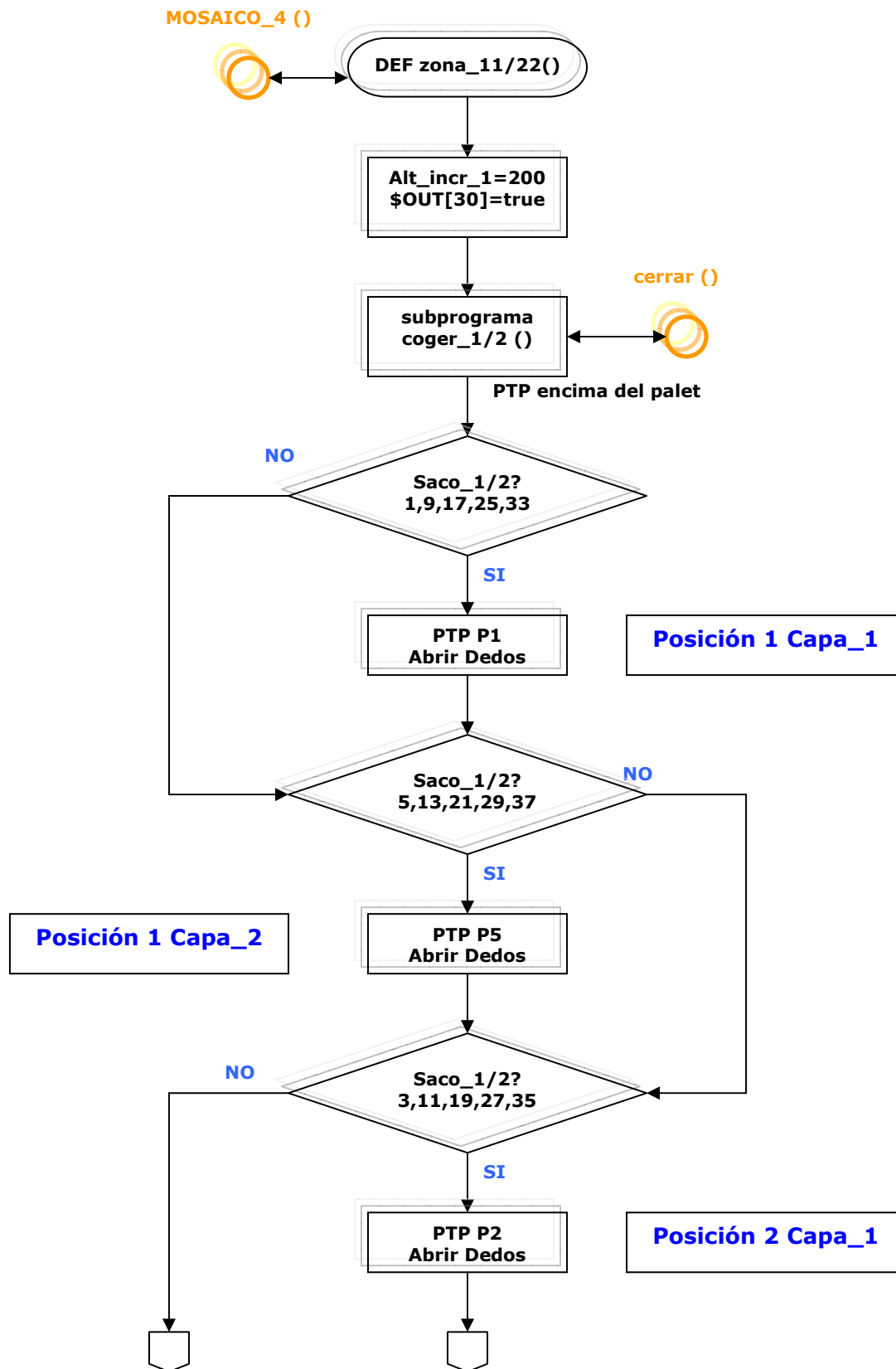
Después de evaluadas las posiciones de saco, se comprobará si el palet esta completo. Esto se hace mediante la igualdad **saco_1/2 == sacos_1/2**, de manera que si el valor de la variable **saco_1/2** coincide con el valor de sacos totales a paletizar **sacos_1/2** se generará la señal de final de palet y se iniciará la variable **saco_1/2 = 0**. Se igualará **Base_data[3] = base_data[1]** o **Base_data[4] = base_data[2]** de manera que el valor de **Z** que se ha ido incrementado adquiere el valor de la variable **Z** de **base_data[1]** o **base_data[2]**, que no se modifica en ningún momento y que guarda las coordenadas iniciales de **base_data[3]** y **base_data[4]** respectivamente en el config.dat.

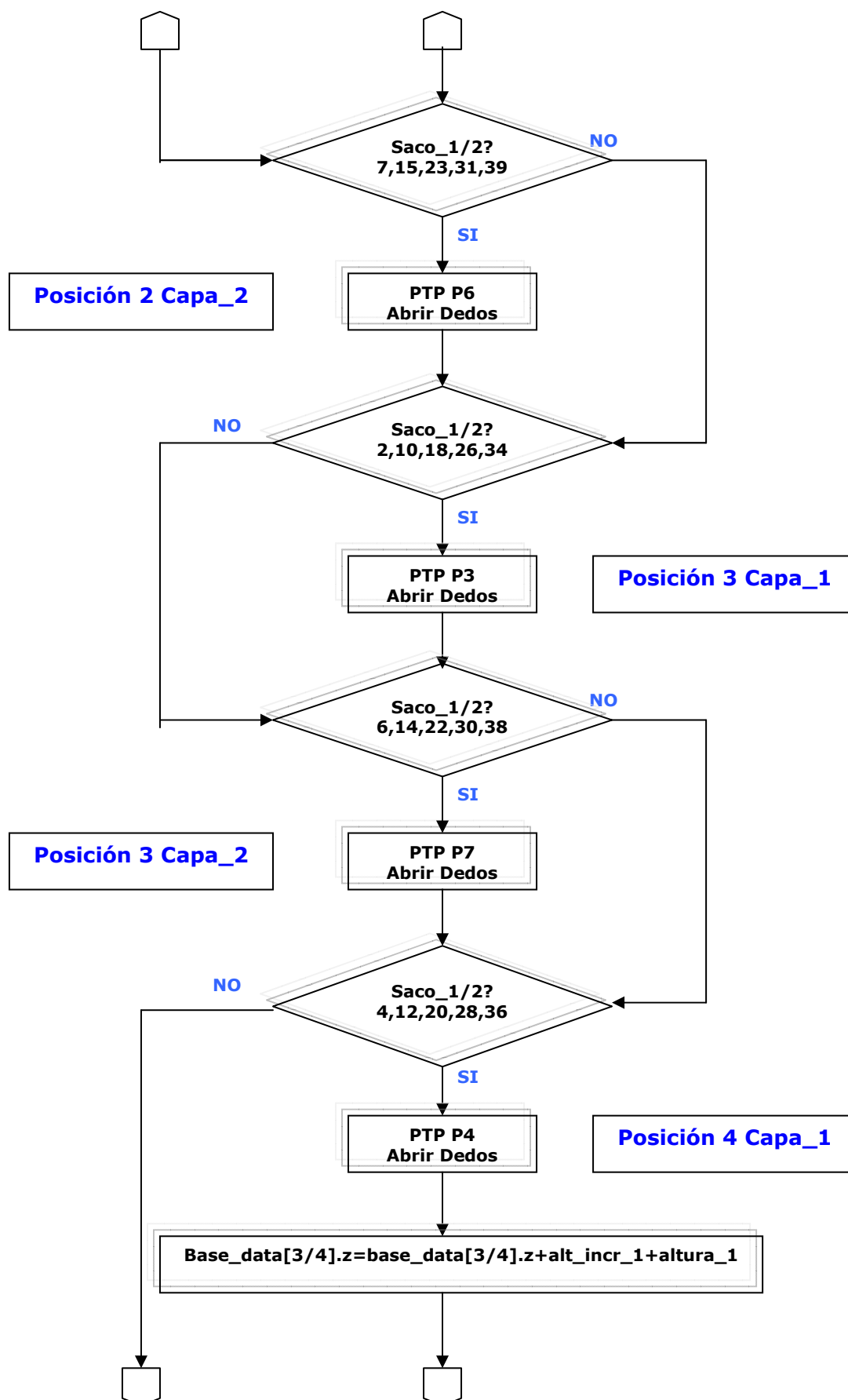
Por último se incrementa el valor de **saco_1/2**, de esta manera en el caso de que se finalice el palet, la variable **saco_1/2** adquirirá el valor mínimo de uno, de manera que para el siguiente palet **saco_1/2 = 1** y entrará en la posición de saco correspondiente.

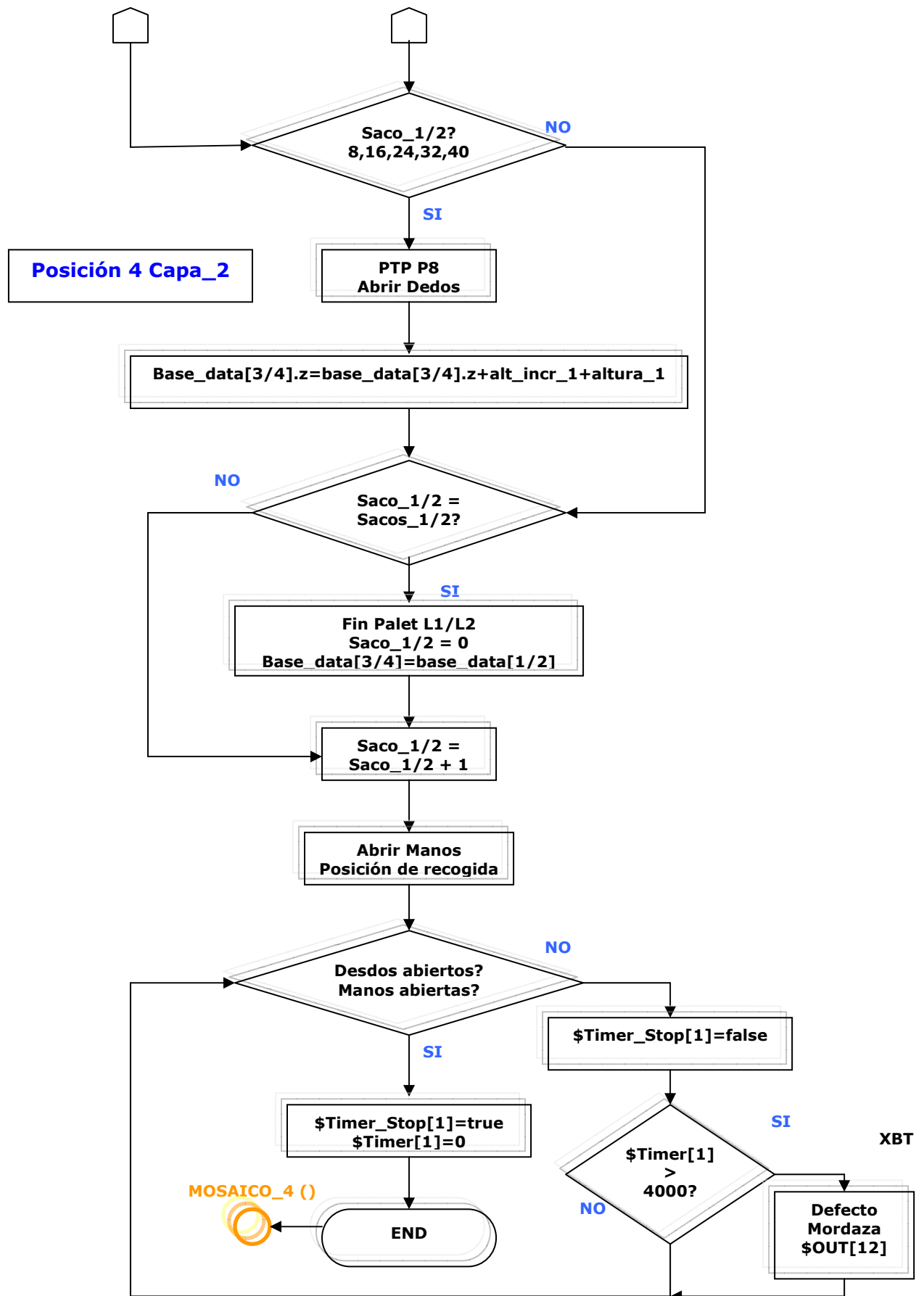
Las siguientes acciones posicionar el robot nuevamente en la zona de recogida de saco, esto se hace a la vez que se abren las manos en trayectoria mediante la función **SYN PULSE**.

El proceso finalizará con el bucle de ejecución para comprobar el estado de reposo de la mordaza, y no se saldrá de este bucle hasta que los dos detectores magnéticos envían las señales correspondientes para el reposo.

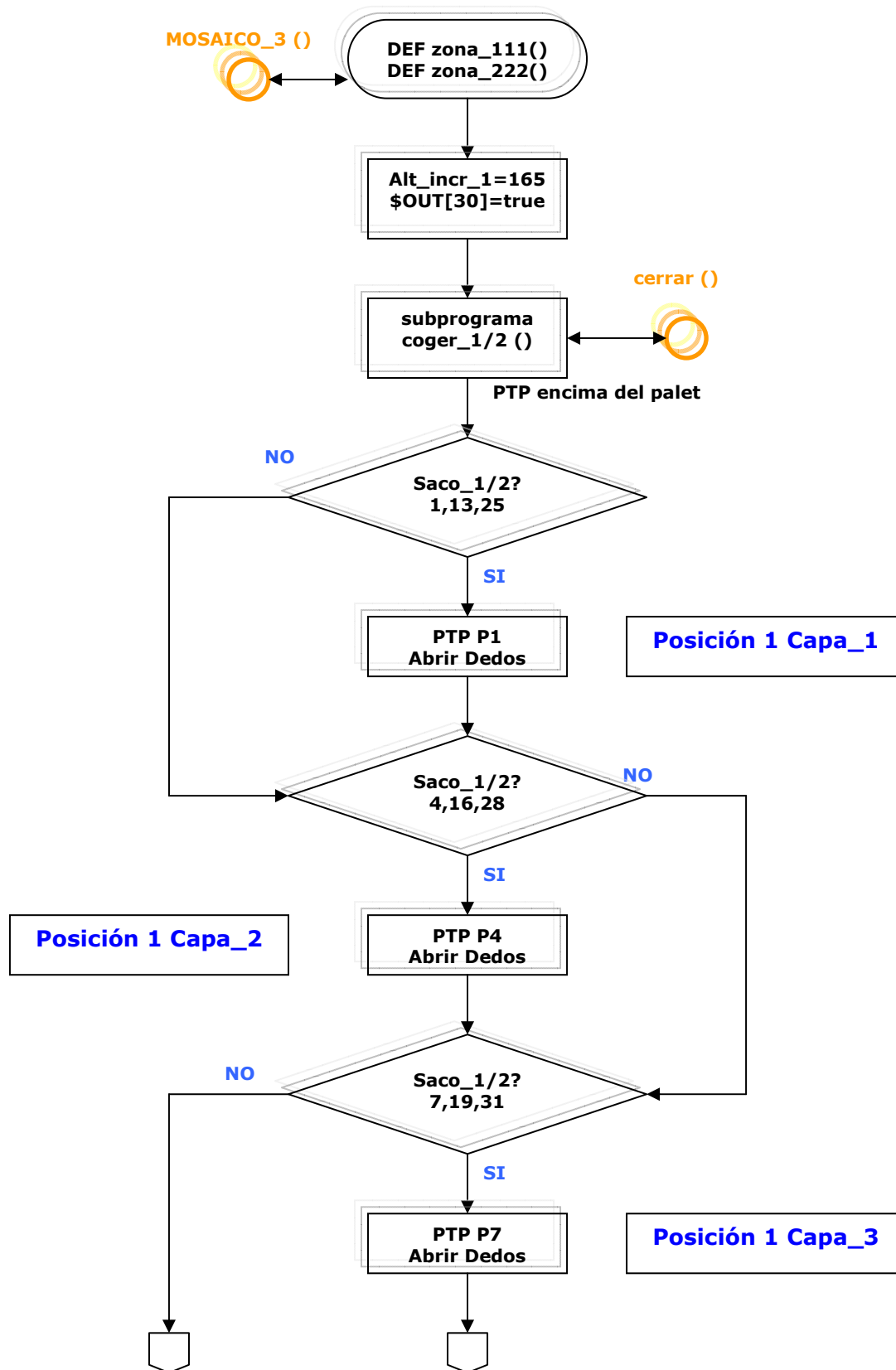
6.2.3.8 Zona_11.src / Zona_22.src

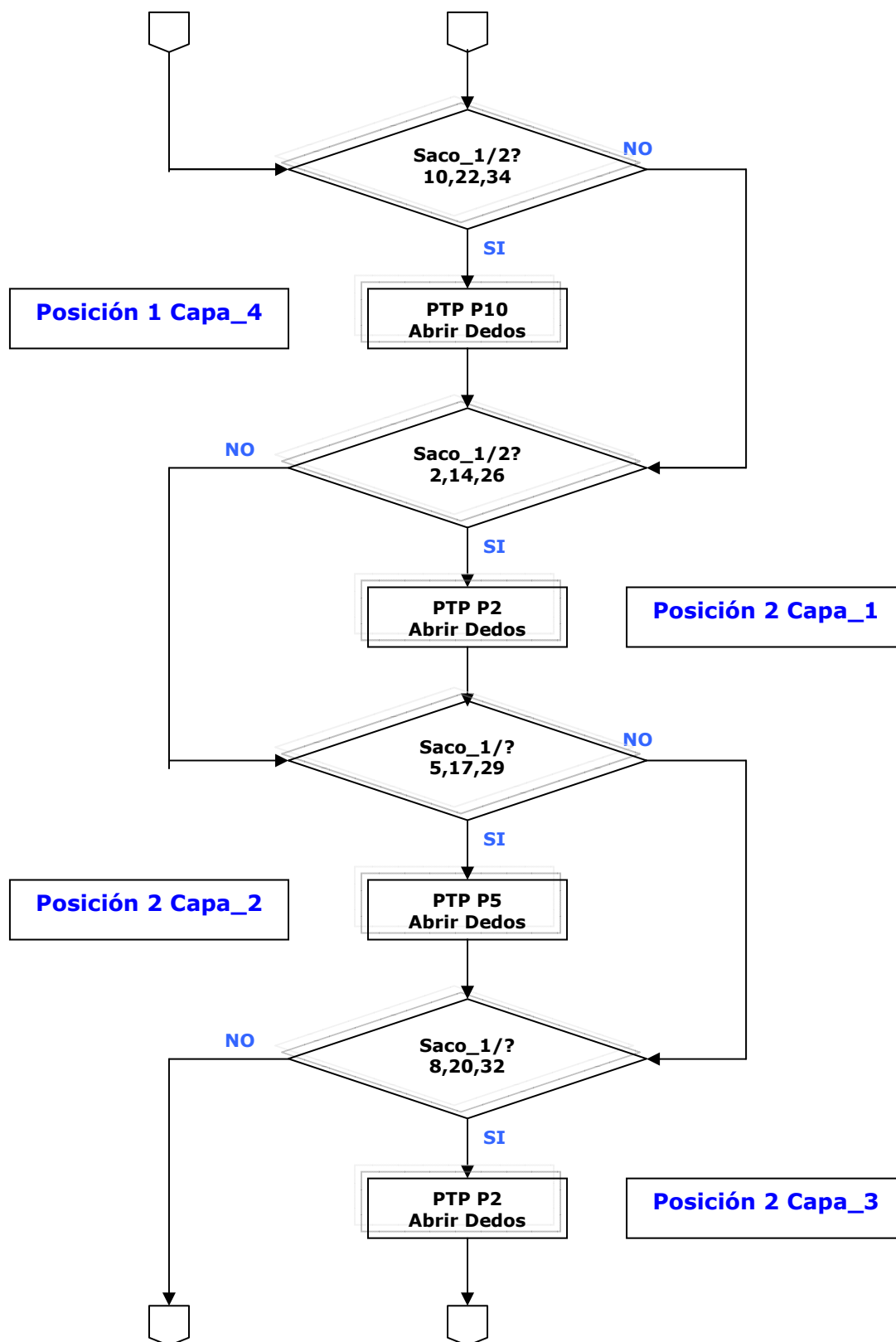


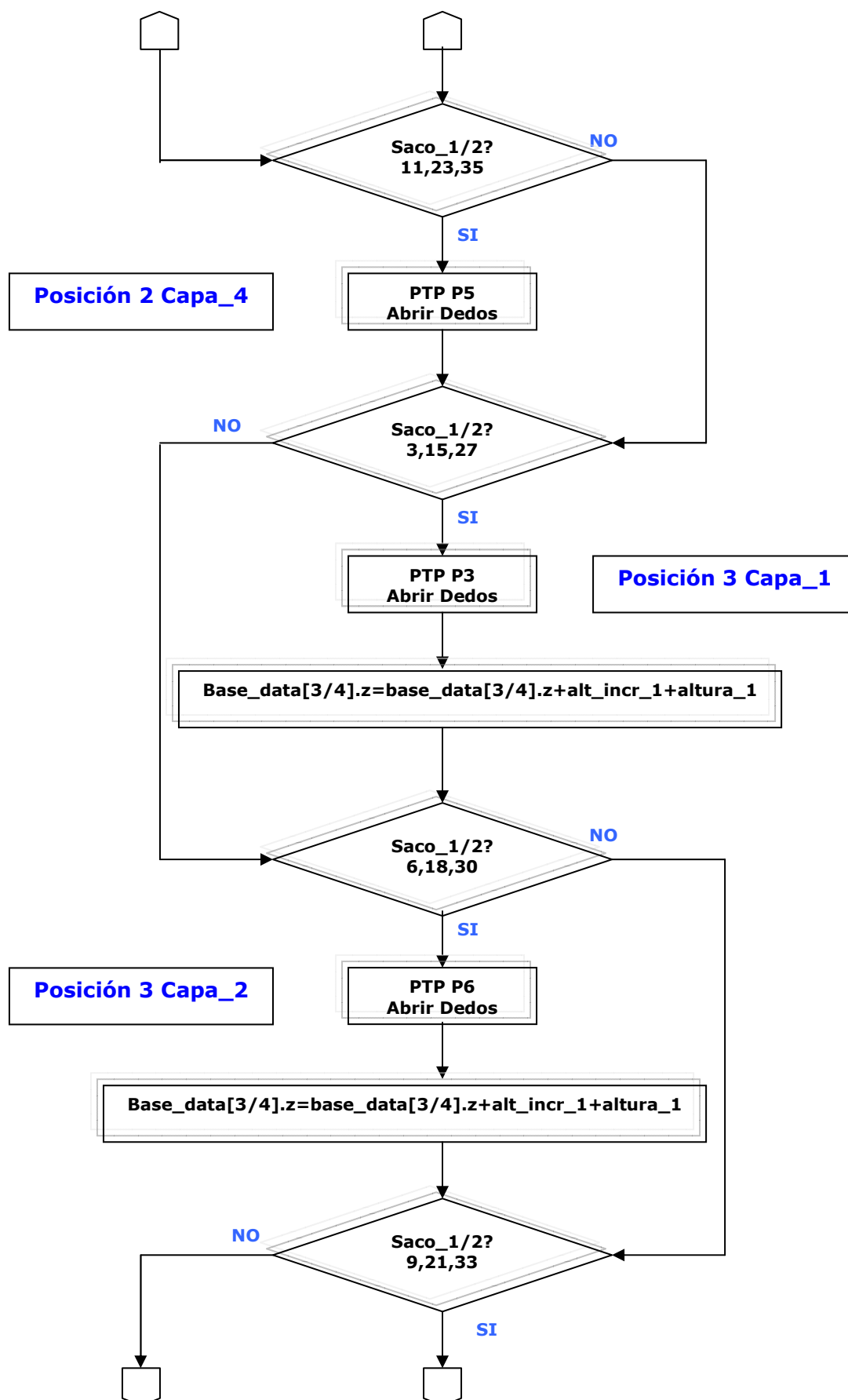


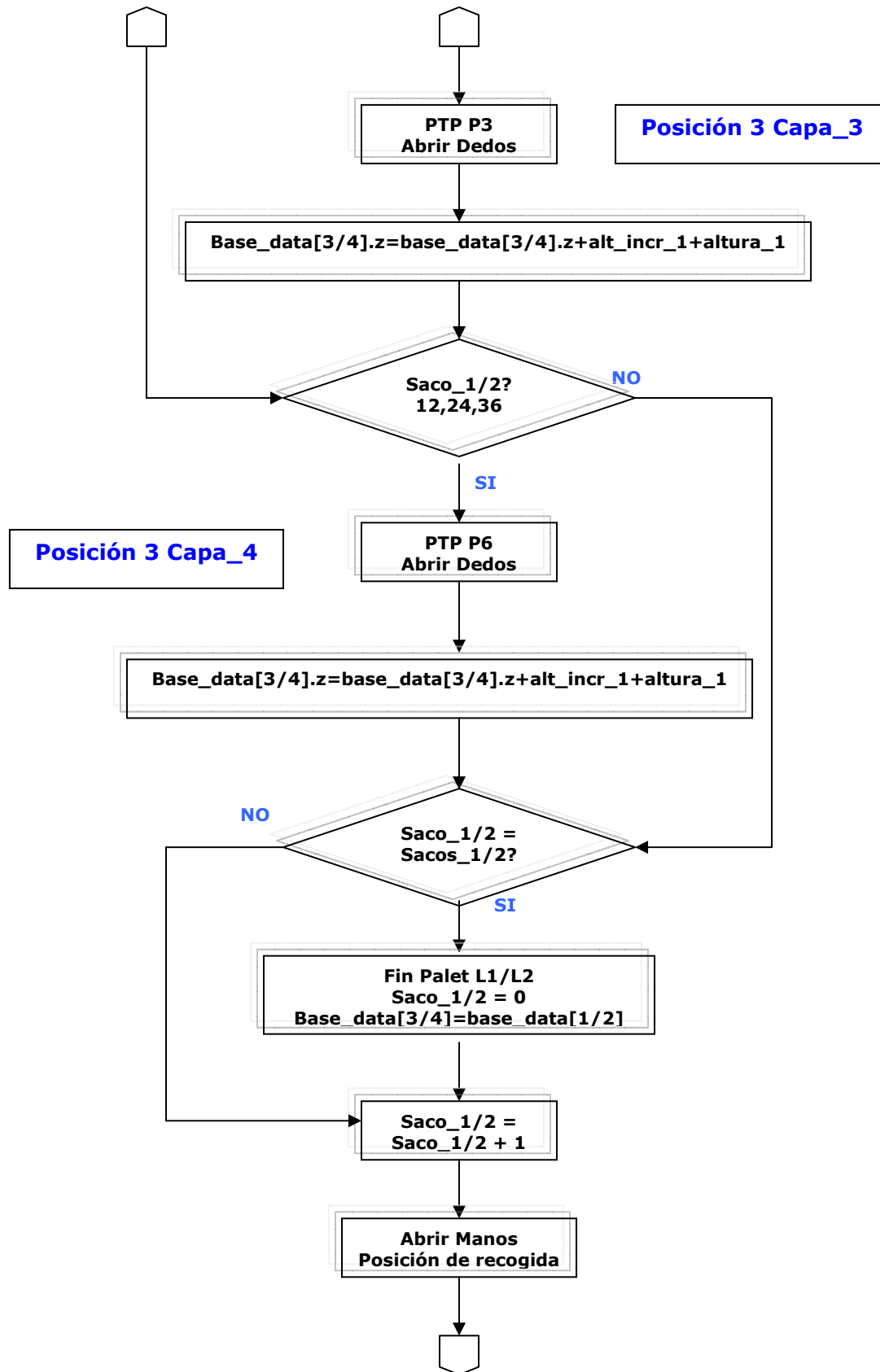


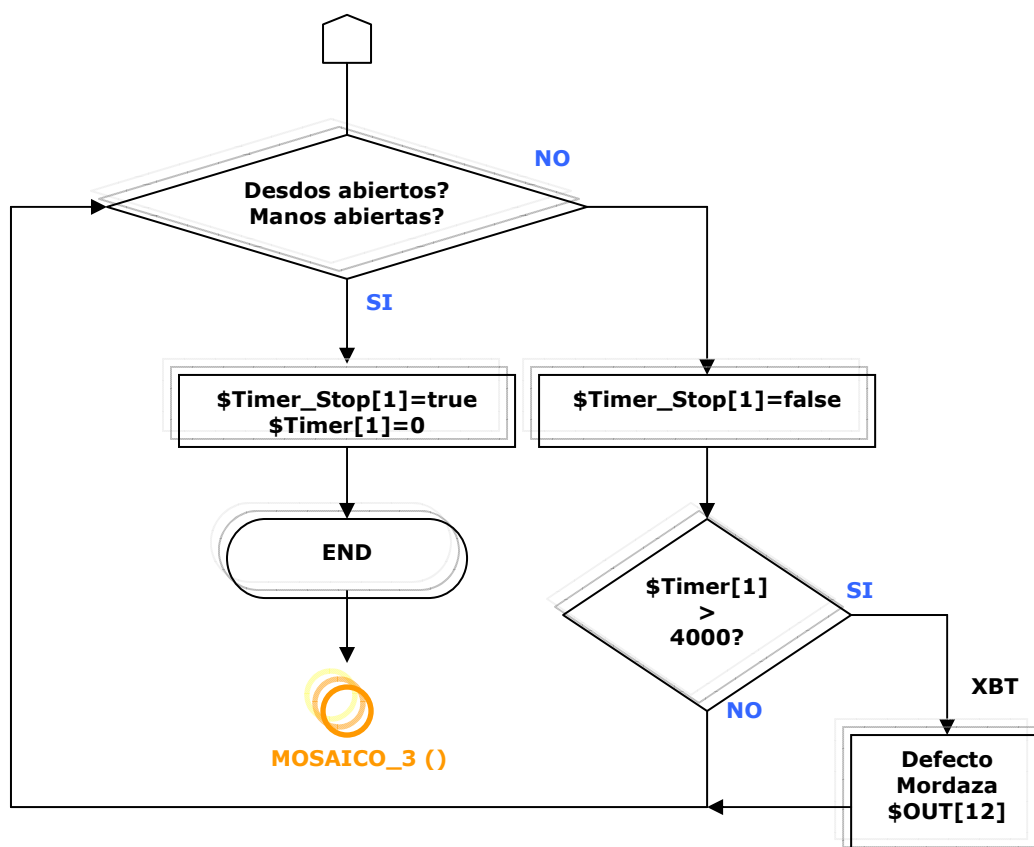
6.2.3.9 Zona_111.src / Zona_222.src



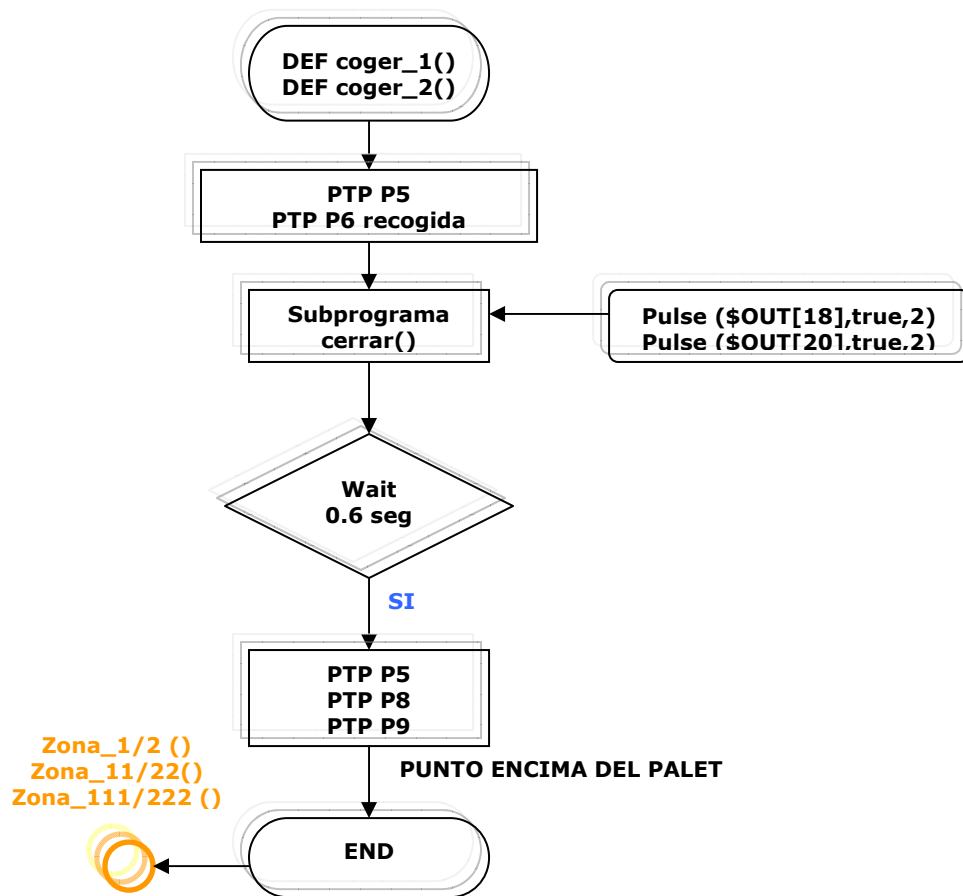










6.2.3.10 Cogrer_1.src / Cogrer_2.src



Los procesos **cogrer_1** y **cogrer_2**, posicionan el robot a la entrada del saco mediante movimientos de velocidad reducida. Una vez posicionado se activaran las salidas para las electroválvulas de cierre de dedos y manos mediante un pulso mantenido de dos segundos, suficiente para la activación de las bobinas de cierre.

Por último se espera 0.6 seg para asegurar el cierre y se posiciona el robot a la altura del palet de la zona correspondiente.

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

6.3 PROGRAMA TSX 3710.

En este apartado del proyecto, detallaremos los procesos llevados a cabo en el programa del autómatas, finalizando así toda la parte software y funcional de la instalación.

En este caso basaremos el siguiente capítulo en una breve descripción de las variables empleadas, pues su entendimiento nos ayudaran conceptualmente para comprender las diferentes secciones que forman el programa.

Distinguimos en este caso las diferentes variables empleadas en el programa; siendo estas, entradas y salidas físicas, marca de memoria tipo bit, palabras de memoria, y por último los temporizadores usados, aunque estas tres últimas clases estarán referenciadas a la explicación lógica de cada una de las entradas y salidas.

6.3.1 Entradas digitales:

En el autómatas distinguimos tres módulos de estradas de 16 bit's conectados por medio de los módulos **10BO** integrados en el armario eléctrico. Cada una de las entradas y salidas se describe con su nomenclatura habitual, junto con nombre de la misma, que aparecerá en el programa y el comentario descriptivo.

- **%I1.0-> Di_boton_paro-> Botón paro instalación.**

Esta entrada corresponde con el pulsador de paro del armario eléctrico, empleado en la sección de marcha general, detectaremos mediante flanco de bajada su accionamiento para desactivar la marca de memoria **%M1** (marcha general).



- **%I1.1-> Di_boton_marcha -> Botón marcha instalación.**

Como en el caso anterior, este pulsador situado en el exterior del armario eléctrico activará mediante flanco positivo la marca de marcha general **%M1**, y una vez puesta en marcha la periferia, mediante detección de flanco de bajada daremos marcha al robot mediante la marca de memoria **%M4**.

- **%I1.3-> Di_boton_reset_alarmas-> Botón reset alarmas.**

Este pulsador corresponde con el rearme de la instalación, y este será usado para rearmar el módulo de seguridades del armario y la marca de seguridades del robot **%M5** (mediante flanco negativo) una vez no existan mensajes por confirmar en la KCP. También se empleará para rearmar el defecto de mordaza en caso de producirse.

- **%I1.4-> Di_seguridades -> Entrada módulo de seguridades.**
- **%I1.5-> Di_presion_aire-> Entrada presión de aire de la maquina.**
- **%I1.6-> Di_serie_termicos-> Serie de los térmicos de la instalación.**

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

Estas tres señales de entrada, se utilizan conjuntamente para activar las seguridades generales a nivel de programa, declarada a través de la marca **%M0** necesaria para que todo pueda ponerse en marcha.

Individualmente se utilizan para activar el defecto en el Terminal XBT, de manera que visualizaremos el estado de los térmicos, del presostato y del módulo de seguridad con estas tres señales.

La entrada **%I1.4** ;señal llevada a través de unos de los reles de mando que indica el estado activado del módulo de seguridades, se utilizará también para desactivar la marca **%M5** de seguridad general del robot. De manera que se precisará posteriormente rearmar sus accionamientos y confirmar los mensajes de estado del KCP.

- **%I1.9-> Di_presencia_palet_lleno_l1-> Presencia de palet lleno en la línea 1.**
- **%I1.10->Di_presencia_palet_lleno_l2->Presencia de palet lleno en la línea 2.**



Las señales de presencia de palet lleno, nos indican con su estado la existencia de sacos en los palet´s. Esta señal será usada en la sección de permisos, como seguridad para que no se active el permiso de descarga en cada una de las líneas cada vez que ponemos un palet nuevo. De esta manera si hay presencia en el palet cuando se ha producido un cambio de palet, la señal o pulsador de palet nuevo no habilitará el permiso de descarga en la línea hasta que se detecte que se encuentra vacío.

- **%I1.11-> Di_celula_recogida_saco-> Célula de mesa de recogida de saco.**

La señal de recogida de saco, representa la fotocélula de presencia de saco en los rodillos de alimentación. Por lo tanto mediante esta señal se determinará los siguientes estados:

1. Su estado libre, permitirá que la cinta de entrada se encuentre permanentemente en marcha en función de **%M1**.
 2. La señal filtrada de esta célula **%M8** mediante temporizador **%TM2** utilizado en la sección de recogida de saco, desactiva la marcha de los rodillos de alimentación, indicando así la llegada del saco a su posición de recogida.
 3. Su presencia junto con la cinta de entrada, activa la señal de saturación, y por lo tanto el paro de la cinta de entrada de sacos del cliente Piensos O´Couto.
 4. Mediante la detección de flancos positivos, incrementamos el contador de sacos de entrada que podemos visualizar desde el terminal XBT.
- **%I1.12-> Di_entrada_saco_saturacion-> Célula de la cinta de entrada del saco para conseguir saturación.**

Esta señal representa la presencia en la cinta de saturación. Mientras este libre de obstáculo, dicha cinta se encontrará en marcha siempre que lo este la instalación **%M1**. Así mismo si pondrá en marcha la cinta de recogida de saco hasta que **%I1.11** detecte su presencia.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

- **%I1.13-> Di_robot_no_actua_I1-> El robot no paletiza en la línea de paletizado 1.**
- **%I1.14-> Di_robot_no_actua_I2 ->El robot no paletiza en la línea de paletizado 2.**

Mediante los selectores para bloquear el paletizado de las líneas, activamos estas señales de mando. Estas señales se evalúan con lógica negada bloqueando directamente los permisos de descarga, en los palet's. Así mismo y cada vez que se requiera un palet nuevo en la línea, será necesario evaluar esta señal previamente, a la vez que se desactivan las barreras de seguridad, de manera que no pueda iniciarse un palet nuevo hasta que se encuentre en su estado de permiso.

- **%I1.15-> Di_boton_palet_nuvo_I1->Palet nuevo puesto en la zona de paletizado.**
- **%I3.0-> Di_boton_palet_nuevo_I2-> palet nuevo colocado en la zona de paletizado esta vacío**

Las señales de palet nuevo, se accionan también desde el armario eléctrico de control, y será la última condición necesaria para volver a activar el permiso de descarga del palet nuevo. De esta manera el cambio e inicio del palet nuevo se producirá en las siguientes condiciones:

1. La instalación está en marcha. **%M1.**
2. El palet situado se encuentra vacío.
3. El selector de bloqueo de línea se encuentra activado.
4. Hubo notificación previa de palet terminado.
5. El pulsador de palet nuevo es accionado.

De esta manera se activará la condición previa para obtener el permiso de descarga mediante la marca de memoria **%M42.**

El permiso de descarga se producirá nuevamente en cuanto volvamos a habilitar la línea de paletizado mediante el selector.

- **%I3.1-> Di_presencia_rodillo.**

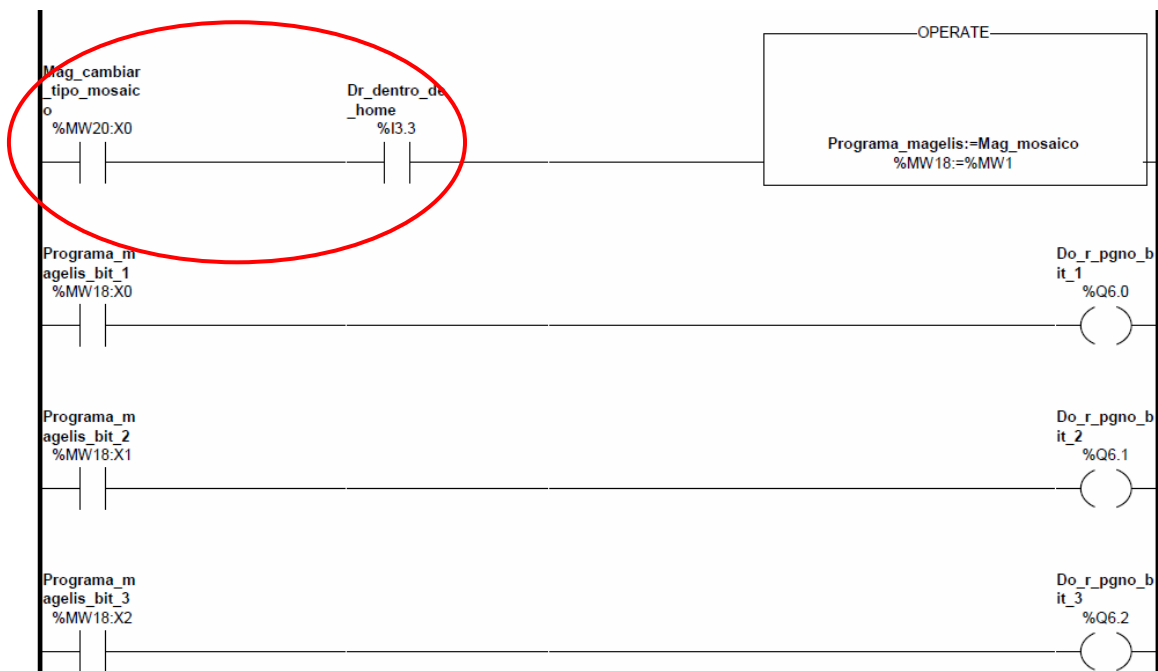
La presencia de rodillo filtrada mediante la memoria **%M9** y a través del temporizador **%TM1** activará la bobina electromagnética para la bajada del rodillo prensa. Este cilindro permanecerá abajo durante el paso del saco, y mientras **%M1** y dicha señal permanezcan activas.

- **%I3.2-> Di_paro_saturacion.**

La señal para el paro de saturación, activará un relé con terminal libre de potencial, que desactivará la cinta del cliente en caso de saturación de sacos. Esta condición se producirá si hay presencia en los rodillos de recogida, y hay presencia en esta señal. De esta manera sin ser necesario evaluar el paro de la cinta de saturación, el saco se detendrá antes del vuelca sacos, no produciendo una nueva entrada a la cinta de saturación.

- **%I3.3-> Dr_dentro_de_home-> Si estoy dentro del HOME se puede cambiar el mosaico**

El PLC es informado del estado dentro de HOME del robot, mediante **%I3.3**. Esta señal será usada por el PLC única y exclusivamente para enviar el valor de programa marcado desde el terminal XBT a través de los bit's PGNO_BIT para el cambio de mosaico. Será la condición necesaria para igualar las palabras de estado **%MW1** y **%MW18** cada vez que enviemos el dato de programa desde el XBT la cual activará la variable **%MW20:X0**.

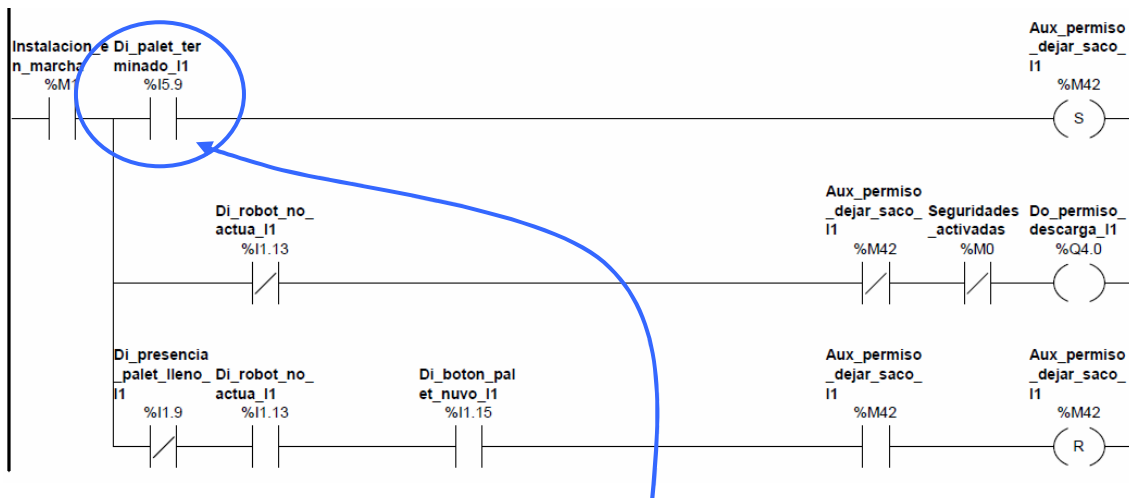


Una vez que el valor del programa es guardado en la variable **%MW118** se enviará a través de los tres primeros bits los datos al robot, mediante los bit's PGNO_BIT.

Dentro de los módulos de entrada 5 y salida 6 nos encontramos la Interface de comunicaciones con el robot, es por ello por lo que se generará un capítulo a parte para estas señales, en donde se detallará expresamente estos procesos, y analizaremos por separado el resto de señales de ambos módulos que no se encuentran integrados en la sección de programa del automático externo.

- **%I5.9-> Di_palet_terminado_l1.**
- **%I5.10-> Di_palet_terminado_l2.**

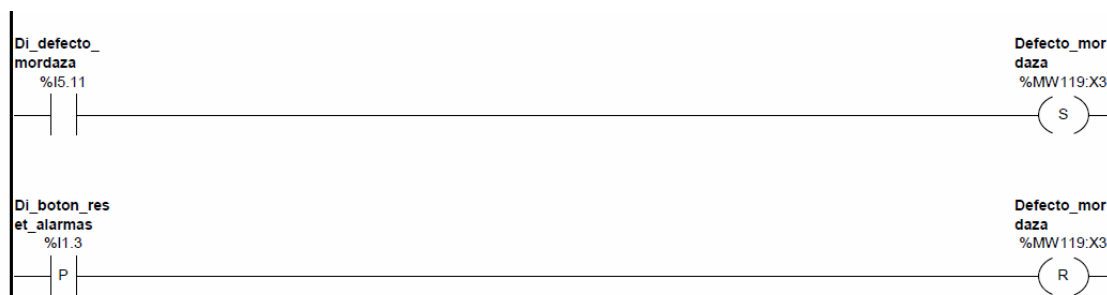
Las señales de palet terminado, se encuentran cableadas directamente desde las salidas beckhoff correspondientes. A través de (**%I5.9**) y (**%I5.10**) el robot indicará al PLC mediante pulsos mantenidos de dos segundos de esta señal, que el palet se ha finalizado en la correspondiente línea. Una vez recibida la señal, se activará en la sección de permisos una marca auxiliar (**%M42**) o (**%M43**) desactivándose así los permisos de descarga hasta que no se desarrolle el procedimiento de palet nuevo mencionado anteriormente.



Para el caso de producirse un **palet terminado** en la línea 1, se desactivará %Q4.0 de permiso de descarga.

- **%I5.11-> Di_defecto_mordaza.**

Señal activada por el KRC, para indicar el defecto de reposo de la mordaza.



Dicho defecto sólo podrá desactivarse única y exclusivamente con el pulsador de reset del armario eléctrico, una vez que el operario haya visualizado el mensaje en el terminal XBT.

- **%Q6.10 -> Do_r_permiso_carga.**

El permiso de descarga será también condición necesaria e indispensable para entrar en las zonas de paletizado Zona_1() o Zona_2(). De esta manera, estando la instalación en marcha, una vez que ha llegado el saco a la presencia de los rodillos de recogida, se activará el permiso de carga siempre y cuando haya algún permiso de descarga.

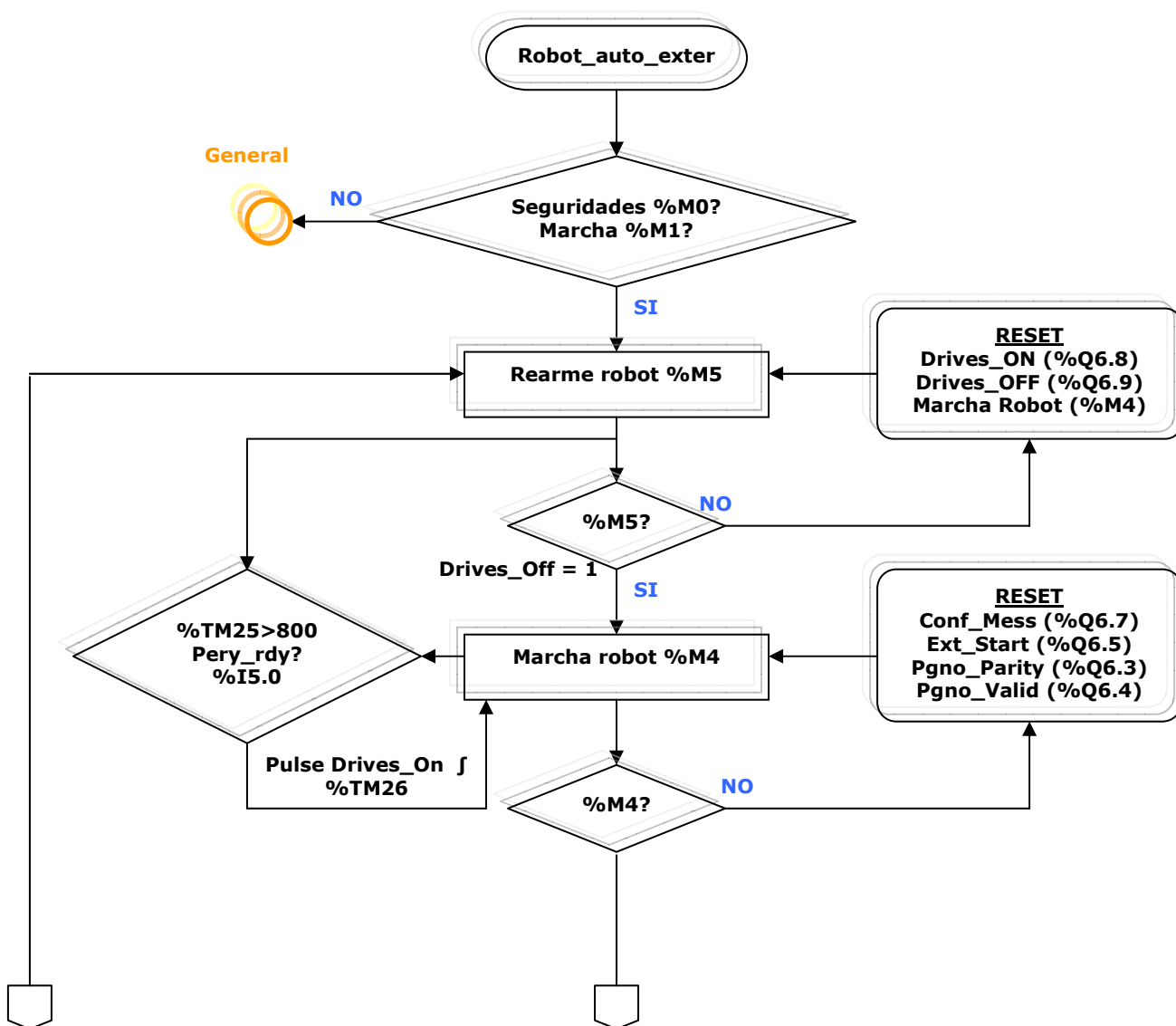


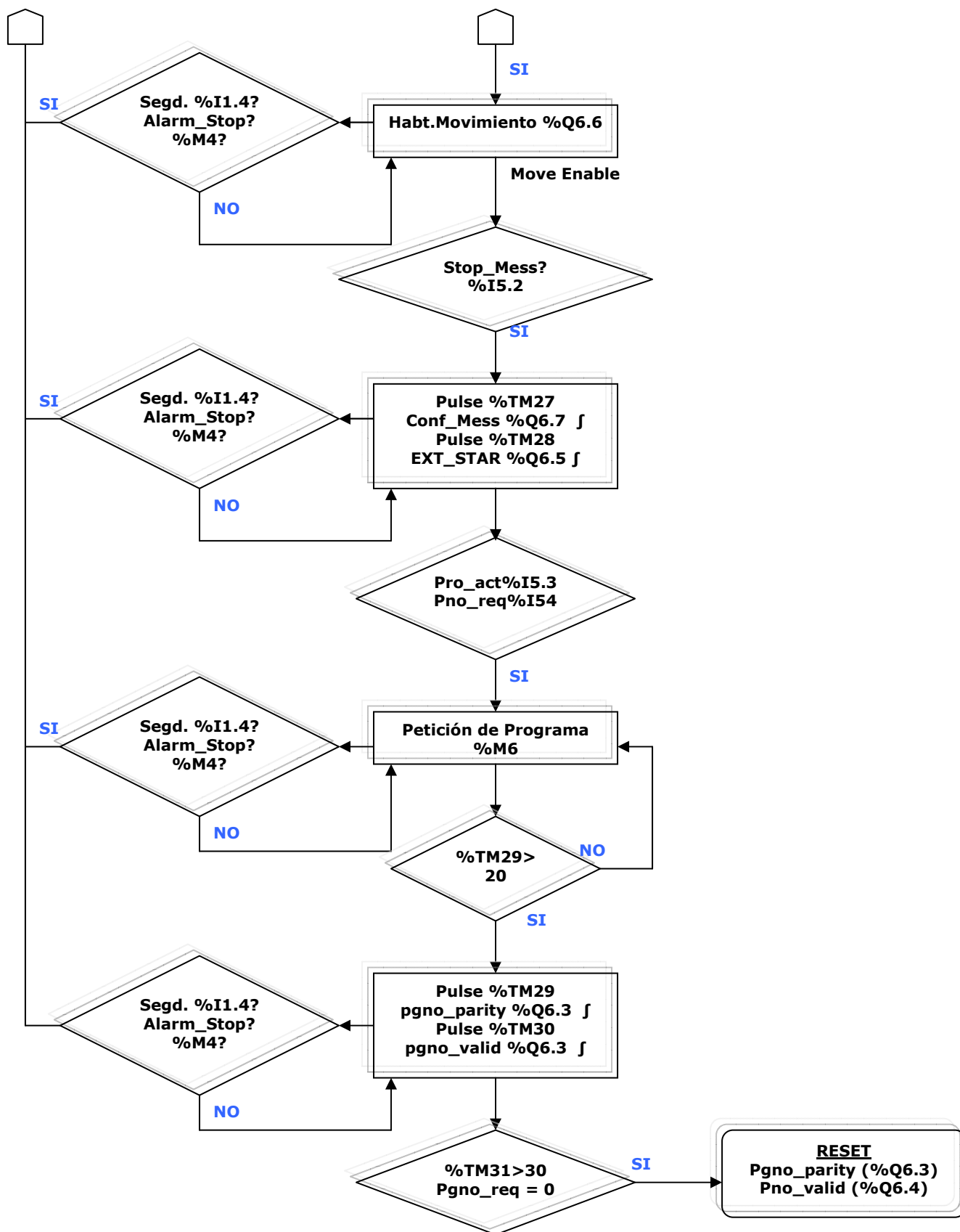
6.3.2 Comunicación con el robot.



Para la comunicación con el robot, se reserva el módulo de entradas número 5 y el módulo de salidas número 6, que componen las señales necesarias para la Interface con el robot mediante automático externo, ver para ello los capítulos correspondientes para esta Interface. La sección dedicada al automático externo se encuentra estructurada según el diagrama de señales adjunto en el capítulo relacionado con este apartado, empezando por la confirmación y activación de los accionamientos del robot, para pasar después a la comunicación y run del programa a ejecutar.

Se representará a continuación el diagrama de estados para esta sección del programa, detallando posteriormente cada uno de sus procesos de acuerdo con el diagrama de señales visto en la sección del automático externo.

6.3.2.1 Diagrama de estados – Manejo del robot.





	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

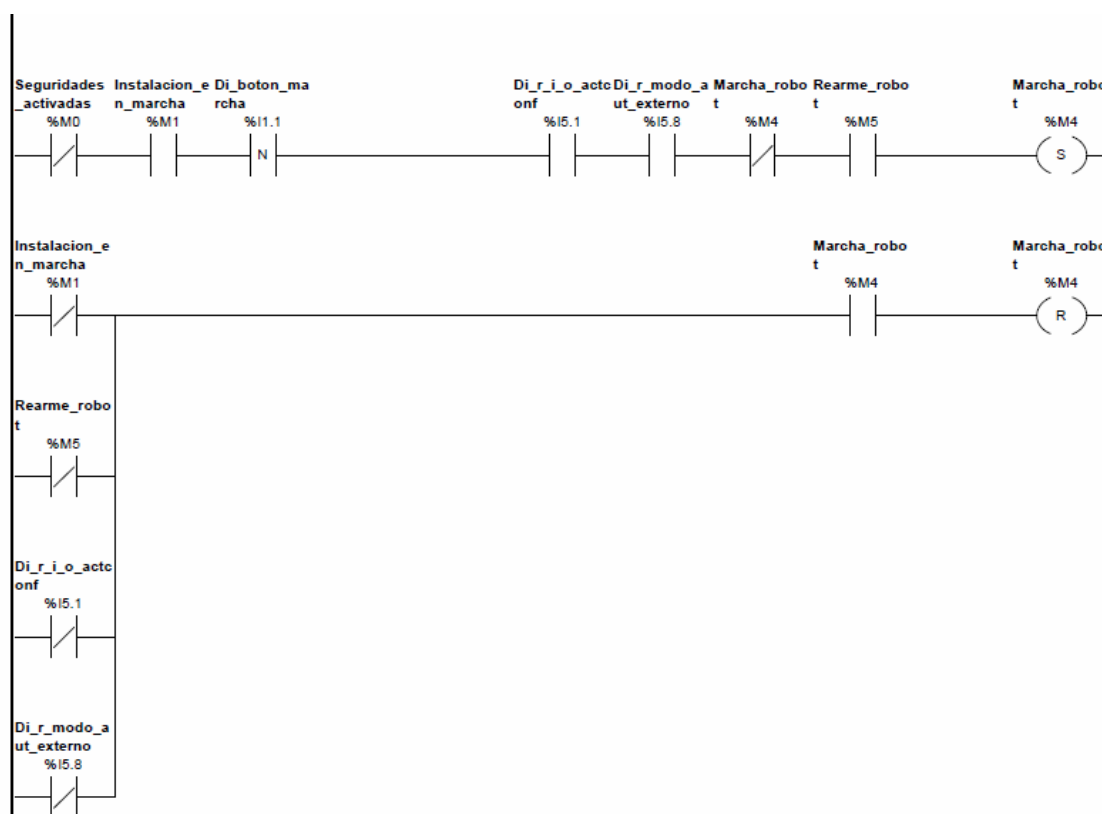
Lo primero que observamos en el diagrama, es la dependencia de las marcas de estado de marcha y seguridad general de la instalación, pues si este apartado no se encuentra OK, no podría funcionar nada.

En la primera parte del programa, se establecerá la marca de memoria que representan el rearme del robot **%M5** será necesario por lo tanto que no exista ninguna alarma en el brazo robot **alarm_stop-> (%I5.6)** para rearmar y poder hacer a continuación la marcha general del robot, **%M4** tendrá que estar activada para que el resto de señales se procesen acuerdo al funcionamiento establecido.

Para la activación de la marcha del robot (**%M4**) se tendrá en cuenta:



1. **%M0->** Las seguridades de la instalación tendrán que estar OK.
2. **%M1->** La marcha general tendrá que estar activada.
3. **%I5.8->modo_aut_externo->** El modo de funcionamiento del robot tendrá que estar establecido en automático externo.
4. **%M5->** Las seguridades del robot tendrán que estar activadas.

Cumplidas estas condiciones, mediante el flanco de bajada del pulsador de marcha general **%I1.1**, activaremos la marcha del robot (**%M4**).



Para la activación y rearme del robot (**%M5**) tendrán que cumplirse las siguientes condiciones:

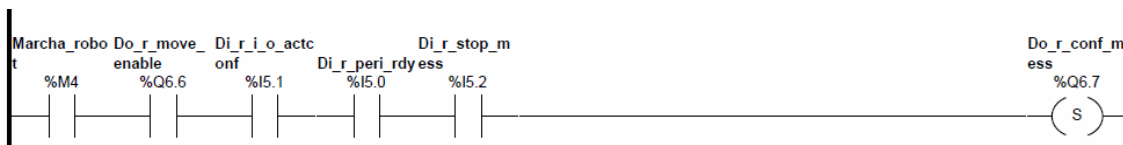
1. **%I5.6->alarm_stop->** No deben existir eventos de alarmas ni mensajes por confirmar en el KRC.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

2. %I1.4->Modulo de seguridad-> El módulo de seguridad del armario de control debe encontrarse activado.

Si **%M5** no se encuentra activado, no se podrán, rearmar los accionamientos del robot (**%Q6.9 Drives_off** y **%Q6.8 Drives_On**), y en caso de no encontrarse activada la marcha del robot **%M4** la Interface de comunicación y transmisión de datos se encontrará desactivada, así como la habilitación del movimiento del robot (**%Q6.6->Move_enable**) que sólo podrá producirse siempre y cuando esta marca se encuentre activada y seleccionado el modo automático externo del selector.

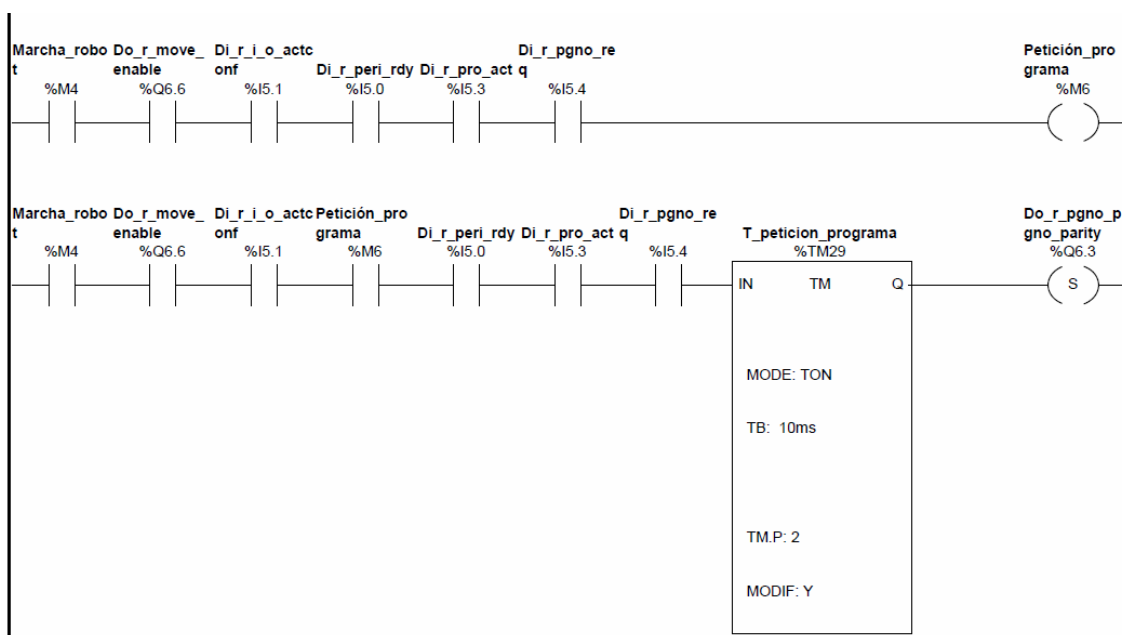
Una vez que los accionamientos del robot están reestablecidos y el movimiento habilitado, será preciso la confirmación de los mensajes de advertencia que el KRC transmite a través de la señal **Stop_Mess**, para lo que se emplea la señal de Interface (**%Q6.7->Conf_Mess**). Si hay mensajes por confirmar (**%I5.2**) estos serán cancelados en la KCP, siempre y cuando la marcha y accionamientos del robot se encuentren activos, para lo que no deberá de haber ninguna seguridad activada.



Una vez el KRC indique la no existencia de ningún mensaje a confirmar, a través del temporizador de 20ms (**%TM27**) de reloj resetearemos la señal (**Q6.7-> Conf_Mess**) y arrancaremos el proceso activo **Cell.src** mediante la activación de la señal **EXT_START** (**%Q6.5**) cuyo pulso de duración dependerá del timer **%TM28** y después de la confirmación del KRC mediante la señal **pro_act->** (**%I5.3**) indicando que el proceso se encuentra activo.

La Interface de señales que da comienzo después de **EXT_START -> Pgno_parity->pgno_valid** se activará después de cada paro y marcha para iniciar la ejecución de un nuevo programa según queda representado en el diagrama de señales en el caso de que nos encontráramos dentro de **cell.src**.

En caso de no existir ningún programa de paletizado en ejecución la KRC activará la señal **PGNO_REQ->** (**%I5.4**) de manera que se procesará la señal **PGNO_PARITY->** (**%Q6.3**) al cabo de un tiempo **%TM29** y siempre y cuando se cumplan todas las condiciones de puesta en marcha.



PGNO_PARITY se encuentra temporizada de manera que al cabo de un tiempo de seguridad, y tras la confirmación del KRC mediante la señal **PGNO_REQ (%I5.4)** se activará el consentimiento de lectura del PLC al KRC a través de la señal **PGNO_VALID (%Q6.4)**. Acto seguido se resetearán ambas señales transcurrido un tiempo de seguridad (**%TM31**) para la lectura del PLC.

Cada vez que se ejecuta un paro desactivándose la marca de memoria (**%M4**) se resetean las señales **pgno_parity-> (%Q6.3)**, **pgno_valid-> (%Q6.4)**, **ext_Start-> (%Q6.5)** y **Conf_mess->(%Q6.7)**, de manera que una vez se vuelva a iniciar la marcha, se confirmarán automáticamente los mensajes de parada anteriores, se reestablecerá nuevamente **EXT_START** y si hubiera petición de programa dentro de **cell.src** se volverían a activar las señales PGNO_PARITY y PGNO_VALID, para el nuevo consentimiento de lectura.

6.3.3 Salidas digitales.

En el autómatas distinguimos tres módulos de salidas de 12 bit's conectados por medio de los módulos **10BO** integrados en el armario eléctrico.

- **%Q2.0-> Do_cinta_entrada_saco_l1-> cinta de entrada saco de la línea 1.**
- **%Q2.1-> Do_camino_recogida_saco_l1->camino de recogida saco línea 1.**

Representan las señales a 24V para activar los contactores de potencia que alimentarán los motores de cinta de saturación y rodillos de alimentación correspondientemente.

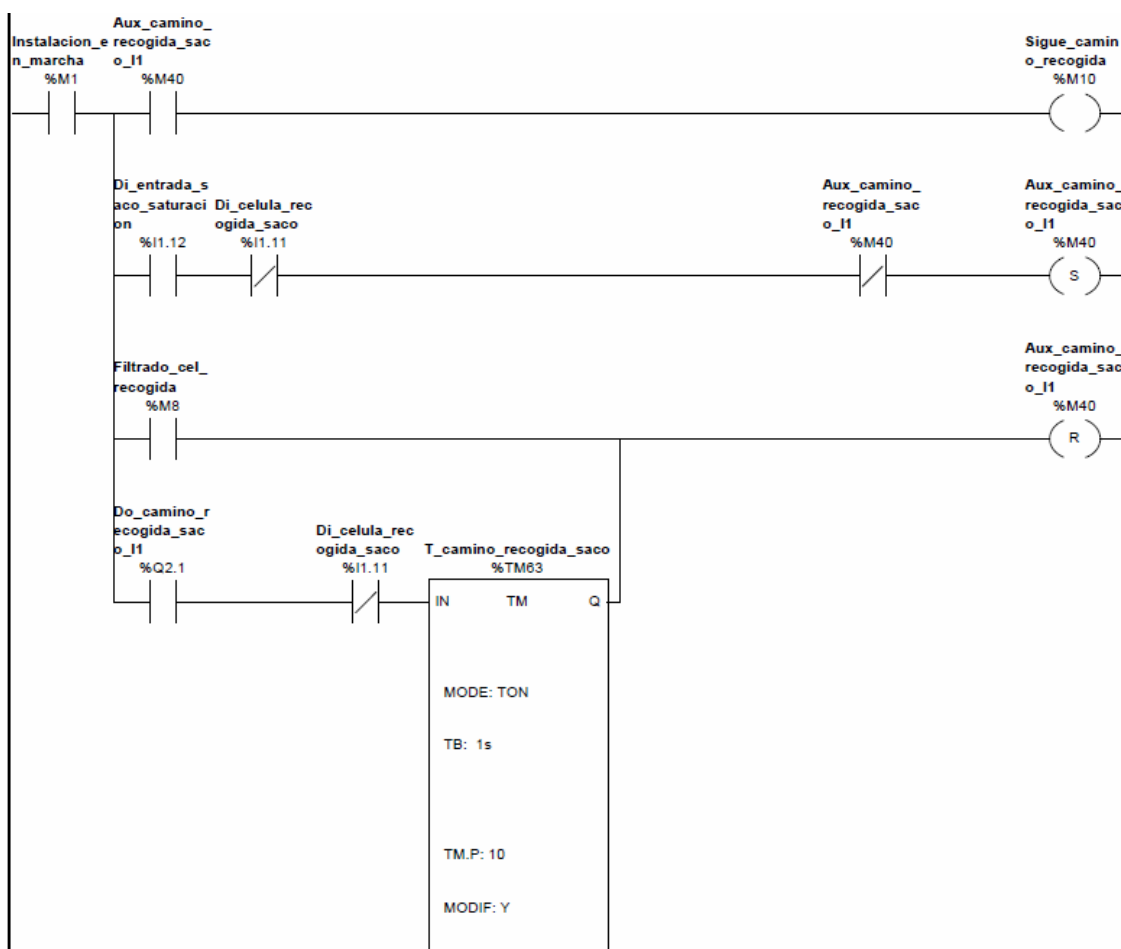


La cinta de saturación estará activa siempre que la instalación se encuentre en marcha, y haya una de las dos fotocélulas de entrada libre. También se activará cuando este seleccionado el movimiento manual desde el XBT, en donde la instalación no deberá encontrarse en marcha.

Los rodillos de alimentación (**%Q2.1**) se activarán a través de la marca auxiliar (**%M10**) que será activada bajo las siguientes condiciones:

1. La instalación tendrá que encontrarse en marcha (**%M1**).
2. Presencia previa en la cinta de saturación (**%I1.12**).
3. No haya presencia en los rodillos de alimentación.

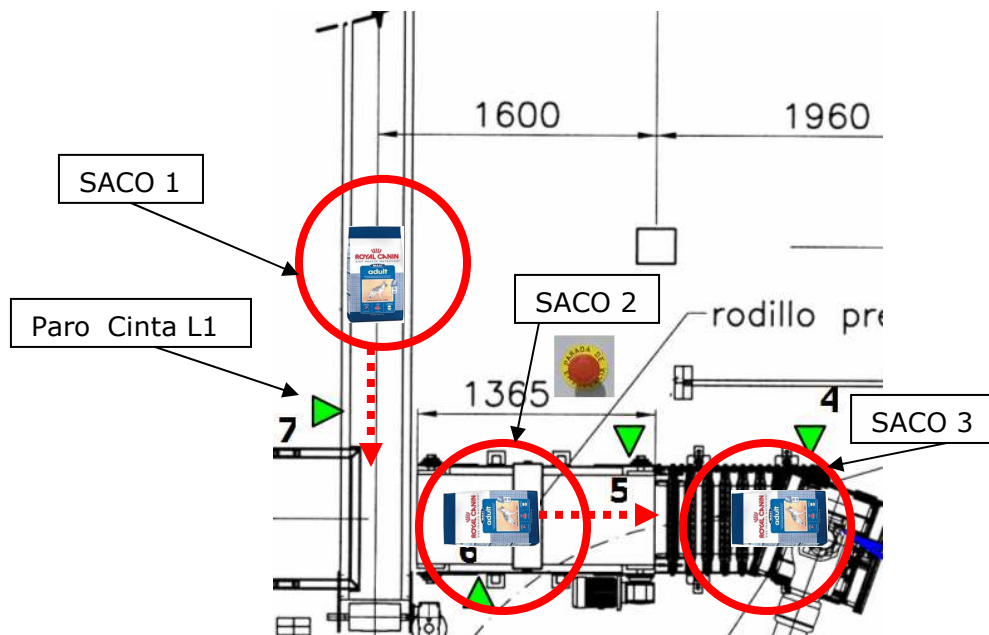
A parte de esto se contempla la posibilidad de una falsa detección en la presencia de la cinta de saturación; es decir, si al cabo de un tiempo (**%TM63**) de 10 segundos de producirse la marcha de los rodillos, no se detecta presencia de saco en la fotocélula de recogida, se desactivará la señal.



- **%Q2.2-> Do_paro_I1-> Paro línea 1.**

Existe una seguridad de saturación que para la línea de entrada del cliente antes del vuelca sacos, para evitar el atascamiento del mismo. La situación se produce cuando habiendo un saco en los rodillos de alimentación se produce una presencia de saco antes del vuelca sacos, contando con que puede existir un saco en la cinta de saturación, no se debería de accionar el vuelca sacos ya que la cinta de saturación estaría parada y el saco se podría quedar atascado. De esta manera a través de esta señal se activará un relé con contacto libre de potencial por el cual se lleva la alimentación al contactor de la cinta de entrada, provocando así su paro hasta que el robot recoja el saco correspondiente dejando libre la presencia.

En la imagen adjunta se observa que si detectamos presencia con el **saco 1** habiendo un **saco 3** en los rodillos de alimentación, la cinta de entrada tendría que pararse puesto que el **saco 2** hará detenerse la cinta de saturación y el **saco 1** podría quedarse atascado si actuase el vuelca sacos.

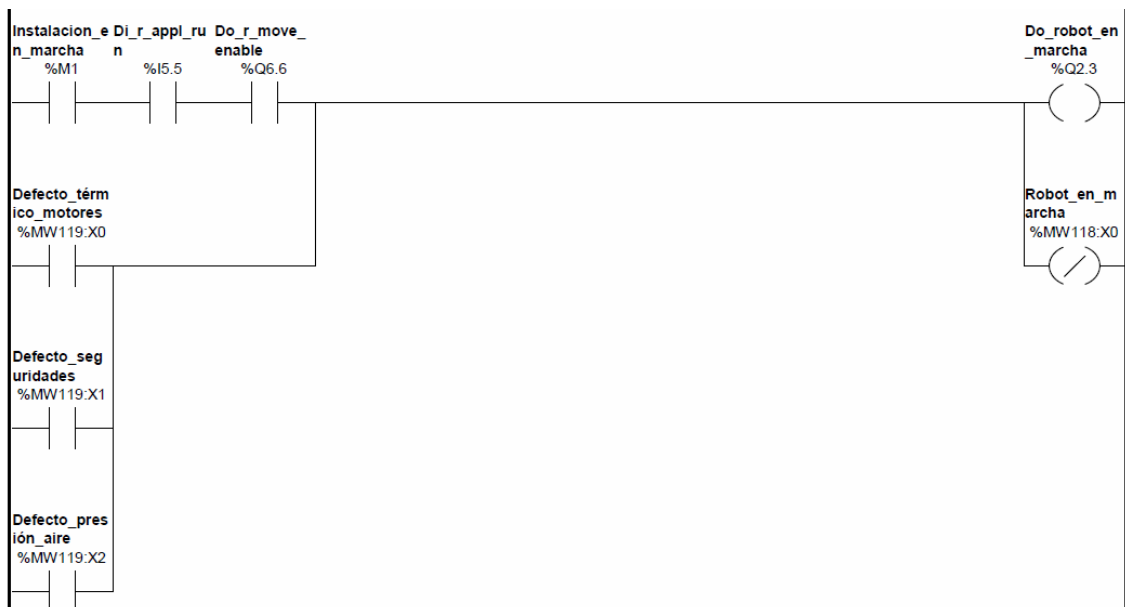




• **%Q2.3-> Do_robot_en_marcha-> Robot en marcha.**

La salida de marcha robot, se emplea para su visualización en el terminal XBT. Así pues dentro de la sección de alarmas programaremos las condiciones que son necesarias para señalar la marcha del robot, que son las siguientes:

1. Marcha general **%M1.**
2. Existencia de programa de paletizado en ejecución **appl_run-> (%I5.5).**
3. Movimiento habilitado **%Q6.6**

Dicho mensaje se visualizara siempre y cuando no se produzca ningún defecto de seguridad con mayor prioridad.



	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

- **%Q2.4-> Do_bajar_rodillo-> Rodillo para prensar el saco.**

Los rodillos actuarán siempre y cuando la instalación se encuentre en marcha y mientras se produzca presencia en **%I3.1->Presencia de rodillo**. La salida **%Q2.4->Bajar_rodillo** representa la activación de la bobina electromagnética del cilindro prensa.

- **%Q4.0-> Do_permiso_descarga_I1.**
- **%Q4.1-> Do_permiso_descarga_I2.**

Los permisos de descarga de la línea de paletizado, son condiciones directas del robot para entrar en las zonas de paletizado. Así mismo si alguna de estas señales no se encuentra activa, no se enviará al KRC el permiso de descarga comentado anteriormente. Las condiciones necesarias para que el PLC active dicha señal son las siguientes:

1. Marcha y seguridades generales activas. (**%M0**) y (**%M1**).
2. Confirmación por parte del KRC de la finalización del palet (**%I5.9/10**).
3. Selector de quitar permiso de paletizado desactivado (**%I.13/14**).
4. Desactivación de la marca auxiliar de permisos **%M42**.

6.3.4 Comunicación con terminal XBT.

Las siguientes señales se encuentran en la sección de comunicaciones, donde se establece el traspaso de datos desde el terminal XBT hacia el robot. Ya se comentó anteriormente cuáles eran el significado de estas señales de entrada directa al robot.

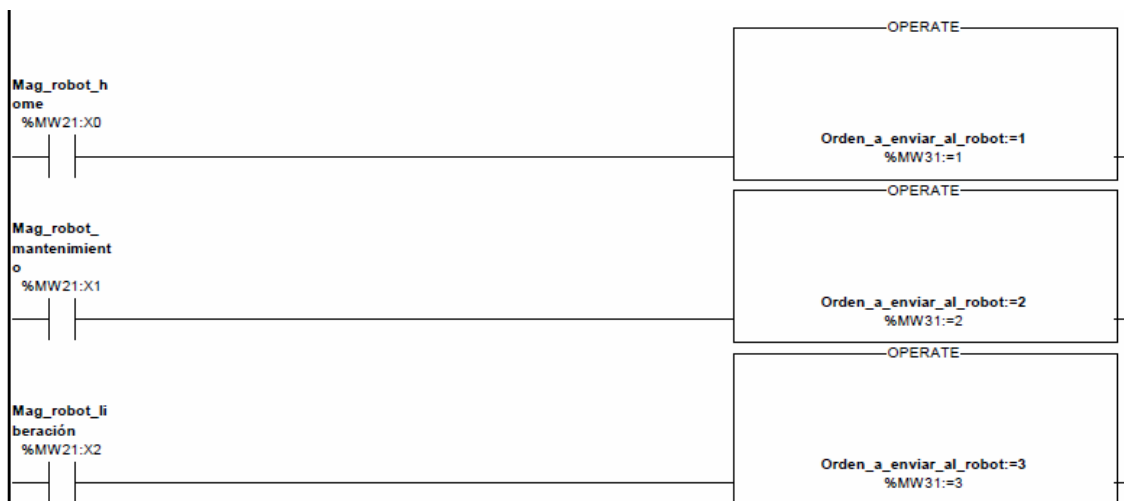
- **%Q4.2-> Do_robot_carga_datos-> ROBOT:Carga datos línea 1.**
- **%Q4.4-> Do_robot_carga_datos_mosaico-> ROBOT: Carga datos de mosaico**
- **%Q4.5-> Do_robot_carga_datos_capas-> ROBOT: Carga datos de numero de capas**
- **%Q4.6-> Do_robot_carga_datos_ordenes-> ROBOT: Carga datos de ordenes**
- **%Q4.7-> Do_robot_carga_datos_incremento->ROBOT: Carga datos de incremento de alturas.**

%Q4.2-> será enviada por el PLC hacia el robot para indicar que se quiere enviar algún dato procesado desde el terminal XBT nº de programa, alturas, etc.

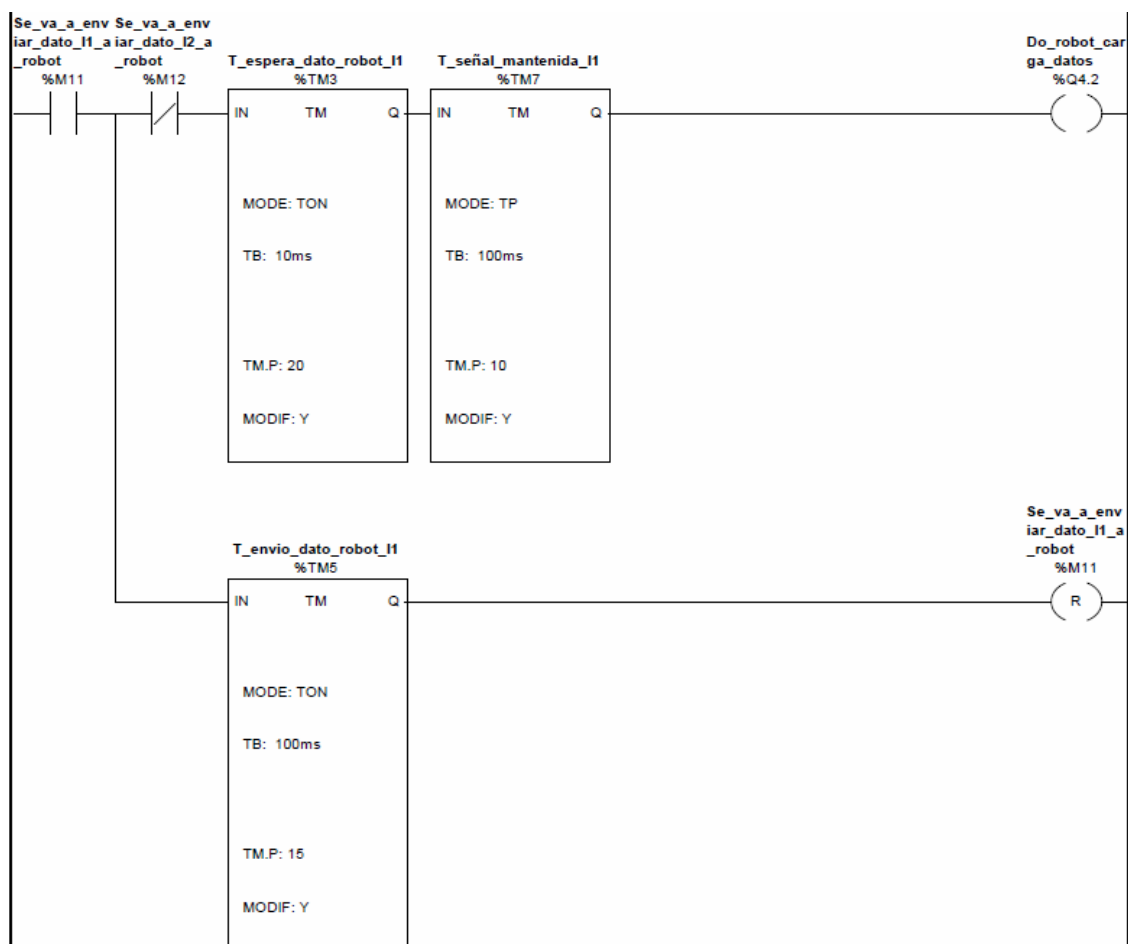


Cada vez que se procesa una orden desde el Terminal XBT se activa la marca de memoria (**%M11**), esto será si ejecutamos algún movimiento manual a través de los bit's de la palabra **%MW20** o si la variable **%MW31** que captura el valor de la orden a través de los bit's de la palabra **%MW21** es distinto de cero. En nuestro caso las únicas órdenes que procesamos desde el XBT son las siguientes:

- **%MW20:X0 -> Confirmación cambio de Mosaico (NO USADO).**
 - **%MW20:X1 -> Confirmación para cambiar Capas.**
 - **%MW20:X2 -> Confirmación para cambiar incrementos.**
 - **%MW20:X6 -> Reset de contadores.**
-
- **%MW21:X0-> Orden de ir a HOME.**
 - **%MW21:X1-> Orden de ir a Mantenimiento.**
 - **%MW21:X2-> Orden de liberación.**
- } **%MW31**



Una vez que la palabra **%MW31** es cargado con algún valor, y a la vez que se activa **%M11**, se activa mediante un pulso de señal la salida **%Q4.2**, este pulso será de una duración de 1 segundo a través de **%TM7** suficiente para que el KRC realice la lectura. Mientras se envía el dato **%M11** volverá a desactivarse mientras que **%M12** permanecerá desactivada puesto que en esta instalación las órdenes procesadas desde el XBT en la línea 1 serán las mismas que en la línea 2.





En el momento que **%M11** es resetada a través de **%TM5** en un tiempo de 1,5 segundos **%Q4.2** será desactivada. De esta manera se le indica al robot que se van a cargar los números de alturas o los números de capas, por lo tanto tendremos lo siguiente:

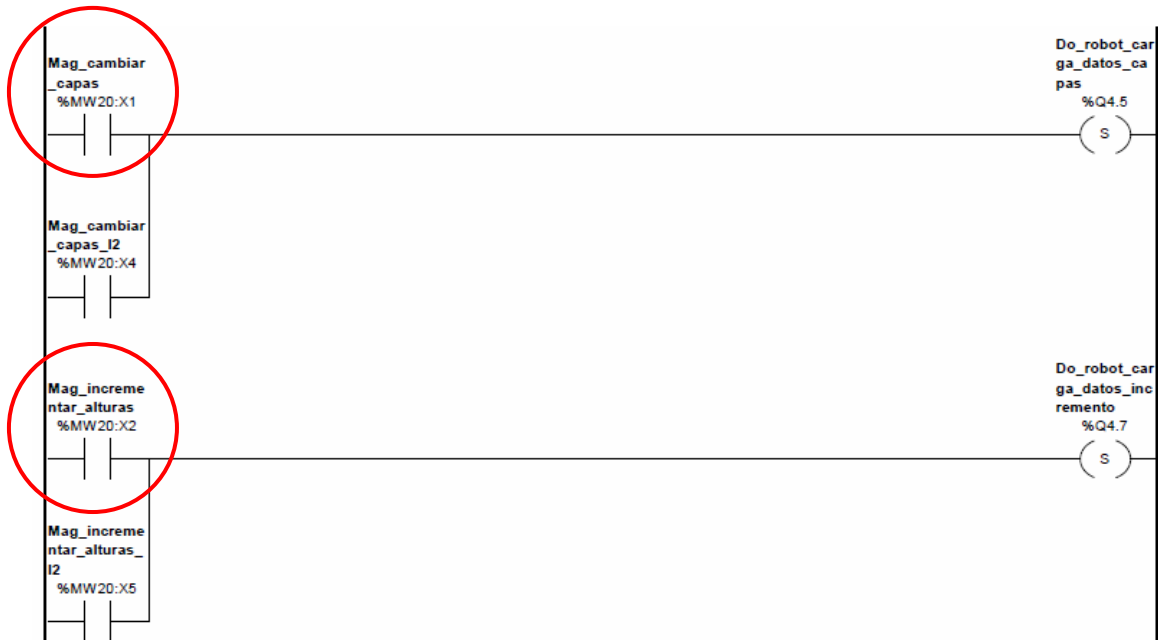
- **%Q4.2->salida PLC->Entrada Directa al robot \$IN[15].**

Se pondrá a nivel **'1'** o **'true'** desde el terminal XBT, para indicarle al robot que se quiere cargar el número de alturas o el número de incrementos. Recordemos que estos valores son recogidos por el programa del robot desde el programa multitarea submit.

Seguidamente será necesario especificar si los datos que se quieren cargar son el número de alturas del palet o el valor del incremento de alturas, puesto que

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

el programa a ejecutar se transmite a través de los bits PGNO_BIT. Para ello si desde el terminal se activa la variable **%MW20:X1->Los datos a cargar serán las capas ->%Q4.5**, y con **%MW20:X2-> Los datos a cargar serán los incrementos-> %Q4.7**.

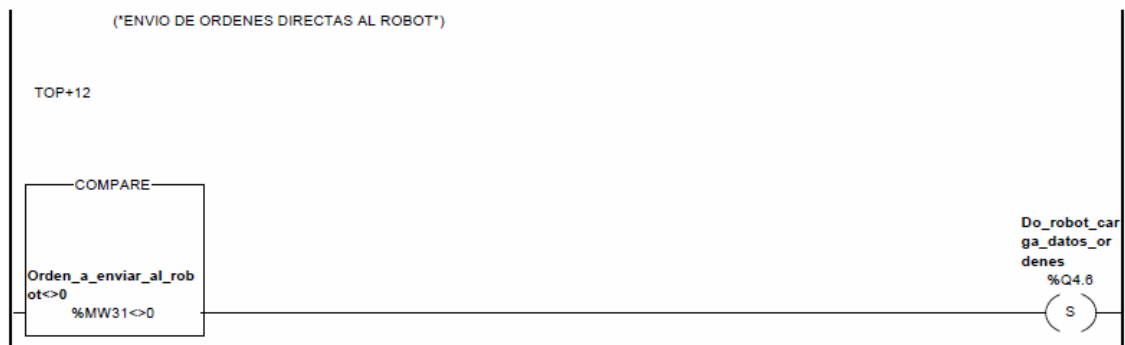


Las variables **%MW20:X4** y **%MW20:X5** no se usan en esta instalación siendo bit's reservados para cuando haya dos líneas a los que se les puede pasar datos diferentes.

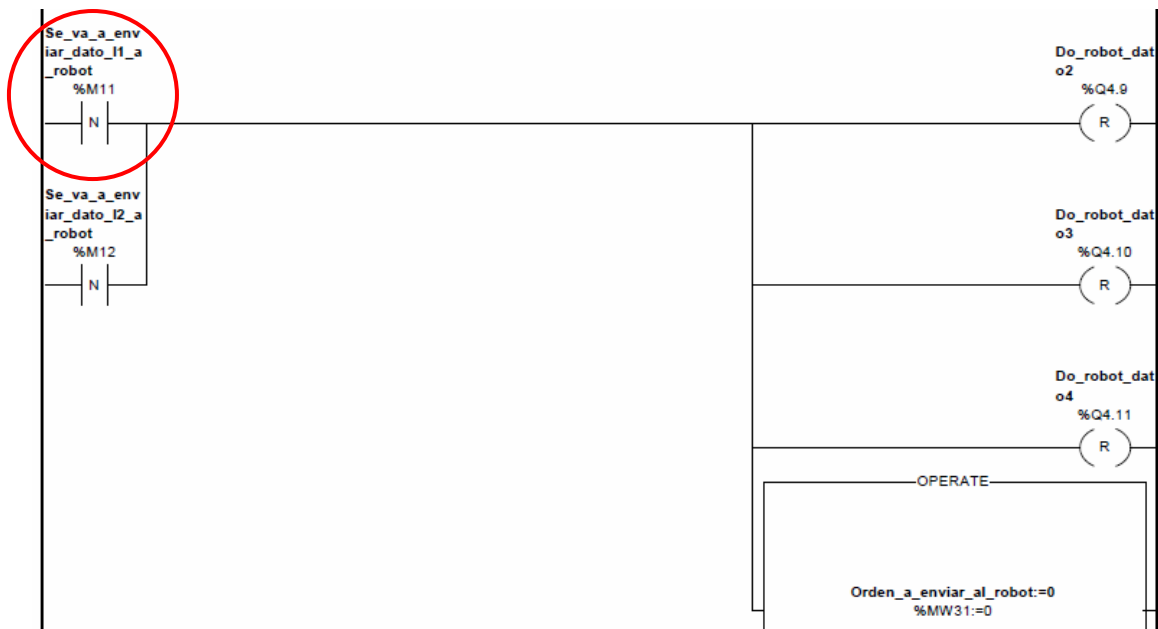
- **%Q4.5->salida PLC->Entrada Directa al robot \$IN[18].**
- **%Q4.7->salida PLC->Entrada Directa al robot \$IN[20].**

Para el caso en el que enviemos el número de mosaico, la salida **%Q4.4-> Salida PLC->No será la entrada Directa al robot \$IN[17]** por lo tanto no será usada aunque su valor cambie de estado desde el terminal XBT, lo que realmente será usado es la variable **%MW1** del terminal que guarda el valor del programa seleccionado. Y que será transmitido una vez el KRC confirme que se encuentra en estado HOME.

Seguidamente será necesario indicar si lo que se quiere hacer es cargar una orden y que tipo de orden, así por ejemplo en el momento que la variable **%MW31** es distinto de cero, también se activará la señal **%Q4.6-> Carga_Datos_de_Ordenes-> Entrada Directa al robot \$IN[19]**. Indicando así al KRC que se va a transmitir una orden mediante la variable **%MW21**.



Por último, las salidas **%Q4.2, %Q4.3, %Q4.4, %Q4.5, %Q4.6, %Q4.7 y %Q4.8 (DATO_1), %Q4.9 (DATO_1), %Q4.10 (DATO_1), %Q4.11 (DATO_1)**, y **%MW31 = 0** serán reseteadas en el momento que **%M11** caiga a nivel bajo nuevamente.

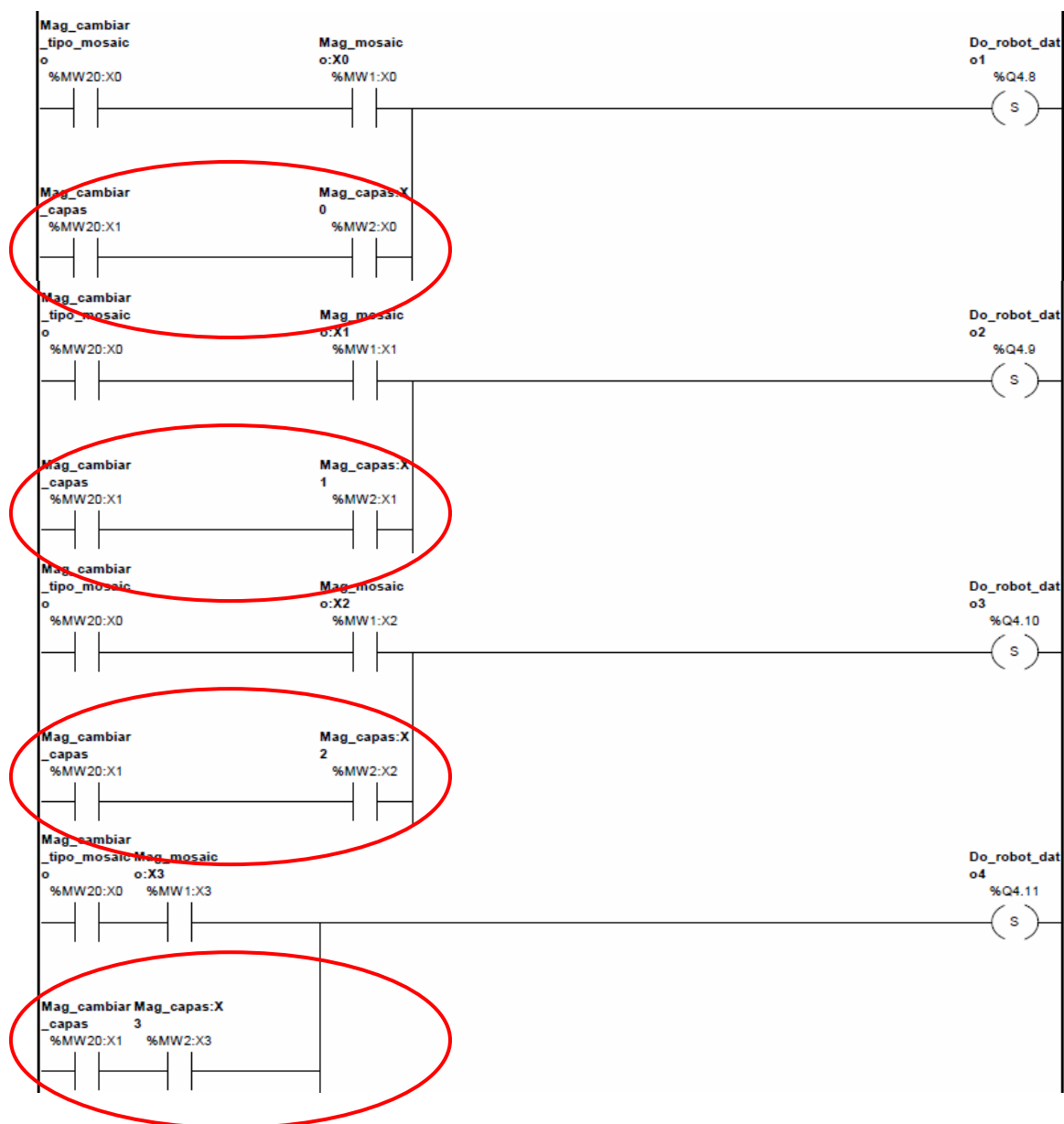


La transmisión de datos se efectúa a través de las siguientes salidas:

- **%Q4.8-> Do_robot_dato1-> ROBOT: Dato 1-> \$IN[21]**
- **%Q4.9-> Do_robot_dato2-> ROBOT: Dato 2 -> \$IN[22]**
- **%Q4.10-> Do_robot_dato3-> ROBOT: Dato 3 -> \$IN[23]**
- **%Q4.11-> Do_robot_dato4-> ROBOT: Dato 4 -> \$IN[24]**

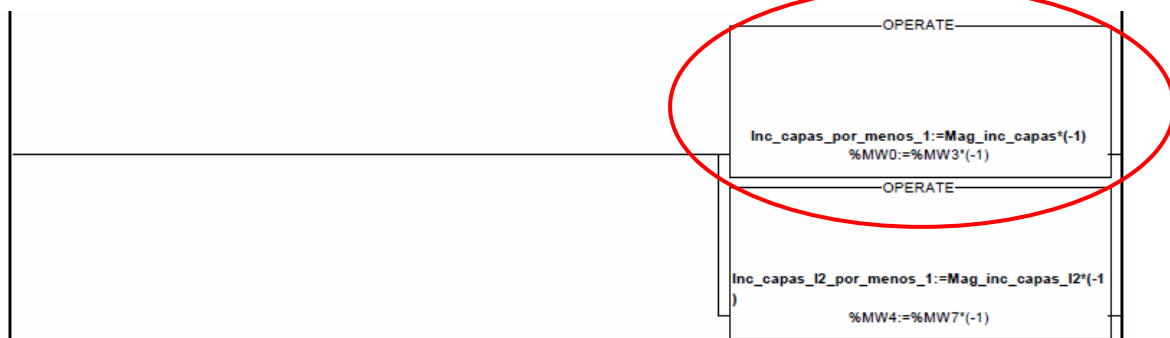
Estos bit's que en el programa del robot forman una señal a través de variables declaradas en el archivo \$Config.dat, se pondrán en estado '1' o '0' dependiendo de las variables anteriores; es decir, si se van a enviar datos se enviará el valor en binario del número de capas o el incremento de alturas, y si se van a ejecutar ordenes se enviarán en binario los valores de la variable **registro_1** declarada en el robot y la cual usamos para entrar en **home()**, **Mantenimiento ()** o provocar la **liberación**.

En el caso que queramos enviar el número de capas o de incrementos, se emplearán como ya se vio en apartado del XBT las variable **%MW2** y **%MW3** respectivamente que guardarán dichos valores, mientras que para activar el envío se emplean los bit's **%MW20:X1** y **%MW20:X2**. que serán las que activen la marca **%M11** en este caso.

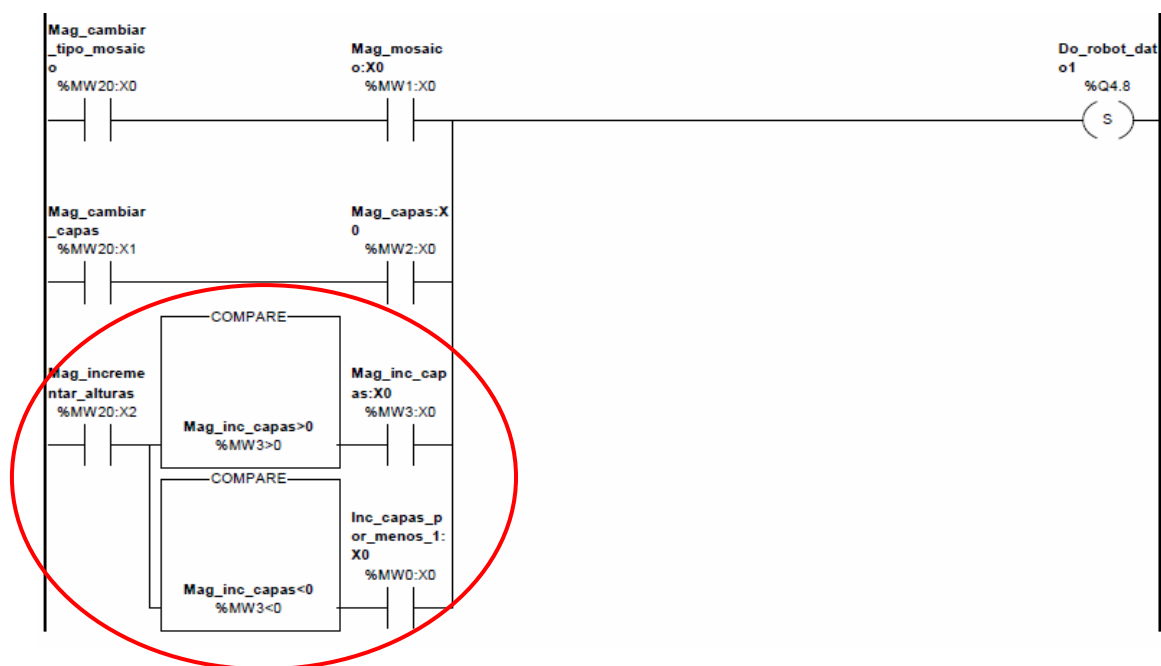


Los bit's de datos transmitirán el dato de capas, si activamos desde el XBT la variable **%MW20:X1** y los bit's de la variable **%MW2** que contienen el valor en binario que seleccionamos desde el Terminal.

Para el caso de los incrementos de alturas, en el caso de que la variable **%MW3** tenga un valor negativo se activará con el mismo valor y signo contrario la variable de signo **%MW0**.



De esta manera **%Q4.8**, **%Q4.9** y **%Q4.10** enviarán el dato a transmitir mientras que **%Q4.11** se activará si el valor a transmitir es negativo.

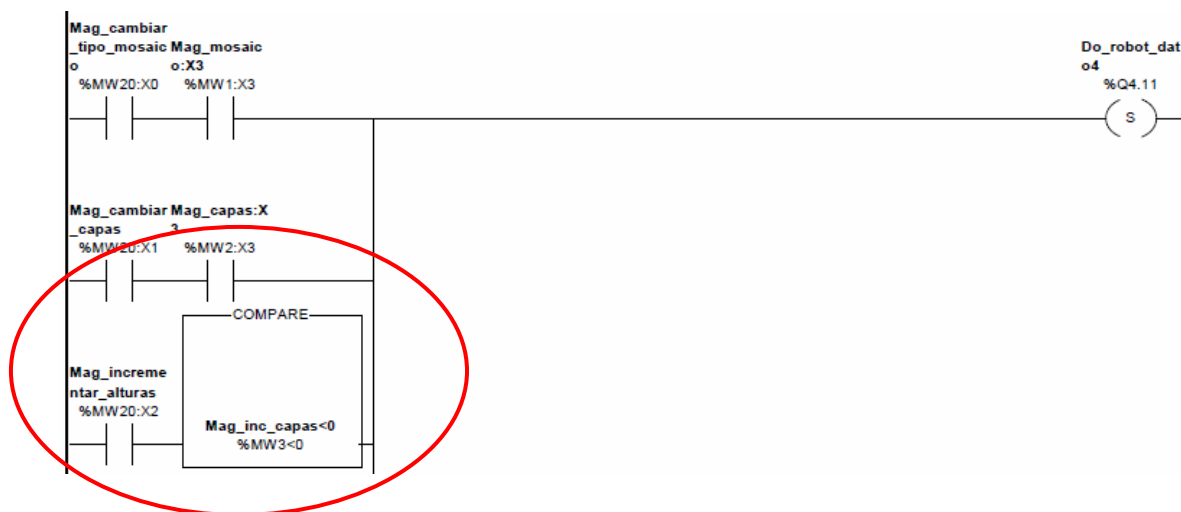


Si activamos el envío de los incrementos mediante **%MW20:X2** tenemos el caso en el que el valor sea positivo o negativo.

- **%MW3 > 0.** {
 - %Q4.8 = %MW3:X0**
 - %Q4.9 = %MW3:X1**
 - %Q4.10 = %MW3:X3**
- **%MW3 < 0.** {
 - %Q4.8 = %MW0:X0**
 - %Q4.9 = %MW0:X1**
 - %Q4.10 = %MW0:X3**

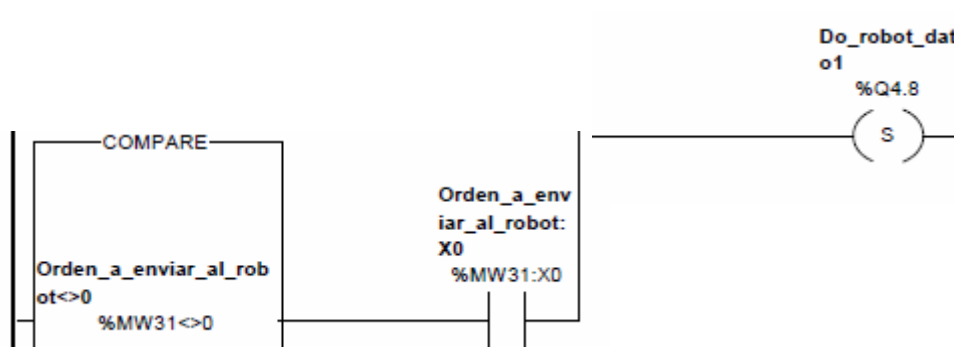
De esta manera siempre se transmitirá el bit positivo tanto si el valor del XBT es positivo como si es negativo.

El bit de signo **%Q4.11** se transmitirá si **%MW3 < 0** lógicamente.



Como último caso nos queda el cambio de la variable **registro_1** del robot cuando lo que transmitimos es un tipo de orden. Las señales de salidas adquirirán los siguientes valores:



- **%MW21:X0 -> Ir a Home -> %MW31 = 1**
- **%MW21:X1 -> Ir a Mantenimiento -> %MW31 = 2**
- **%MW21:X2 -> Liberación -> %MW31 = 3**



Si la orden es distinta de cero, se transmitirá el bit de la orden correspondiente.

	%Q4.11->DATO_4	%Q4.10->DATO_3	%Q4.9->DATO_2	%Q4.8->DATO_1
Registro_1 = 1 HOME	false	false	false	true
Registro_1 = 2 MANTENIMIENTO	false	false	true	false
Registro_1 = 3 LIBERACION	false	false	true	true

Tabla: ORDENES del Robot.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

7 CONCLUSIONES.

"OPTIMIZACION DE UNA INDUSTRIA ALIMENTARIA"

La paletización robotizada como herramienta, ha sustituido la mano de obra aumentando la productividad y la eficiencia. Disminuyó los procedimientos administrativos de recursos humanos y legales que conllevan la rotación de personal; muy necesario siempre para la realización de estos tipos de trabajo.

El principal problema al que se enfrentaba "**Piensos O´ Couto**" radicaba en la necesidad de generar un mayor volumen de suministro para cumplir con sus compromisos de abastecimiento. Esto sin el personal suficiente y las contingencias que se ocasionaban al tratarse de trabajos forzados, limitaba las exigencias productivas de la compañía.

La integración de la instalación de robot paletizado con control automático, garantizó, una mayor producción; sin ser por ello, una herramienta de sustitución del personal que pasarían a realizar trabajos de retirada de palet´s y suministro de material para la realización del ensacado del producto y su transporte hacia la entrada del robot. Todo esto teniendo en cuenta, que el espacio ocupado resultó mínimo, ya que el brazo robotizado se adaptó perfectamente a las limitaciones del recinto, siendo así el modo más adecuado para automatizar la producción del cliente.

Como resultado final, se obtiene un palet totalmente geométrico en las esquinas del palet, aunque esto depende de las características del producto y la homogeneidad con la que el saco llega a la línea de entrada del robot

Se creó en cierto modo, una mayor actividad y más dinamismo a la hora de ejecutar las diferentes operaciones implicadas en los objetivos de la empresa. Pues esto significaba, que disponiendo de los mismos recursos humanos, se llevaban a cabo trabajos de paletizado (automáticos para el robot), a la vez que se transportaba en camiones los palet´s que iban siendo retirados.



Mediante este sistema, se consiguió en caso de necesidad, una disponibilidad del 100%; esto es, 24 horas de trabajo al día durante siete días a la semana con un personal mínimo, siendo capaz de paletizar más 16 toneladas de producto terminado a la hora, lo que hace un total de unas 400 o 500 toneladas al día.

"INSTALACIÓN"

Este tipo de instalaciones, resultan muy versátiles en su ejecución y puesta en marcha. Se requiere poner especial cuidado en la instalación del robot, pues deben medirse correctamente la situación de los elementos y el vallado necesario antes de anclar la base sobre el que se apoyará. Siendo el robot el elemento sobre el que más hincapié hay que hacer en su instalación.

Una vez ubicados todos los elementos, el cableado y las conexiones son sencillas, mediante borneros y conectores.

El cliente ha de proveer de las tomas de alimentación correspondientes para la llegada de potencia a los armarios eléctricos.

	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

En cuanto a inconvenientes, nos encontramos el que una vez situado el robot y ancladas las escuadras de sujeción de los palet's, estos no eran paralelos al sentido positivo en **X+** de la base del robot, por lo que fue necesario crear una base (**base_data[1]** y **base_data[2]**) para cada escuadra que fuera paralelos al eje **X+** de desplazamiento, facilitando así las modificaciones posteriores de las cotas de dejada de saco.

HARDWARE.

El **KR 180-2 PA** por sus características y materiales empleados, resulta ser el robot más idóneo para la paletización. El brazo principal y el brazo paralelo son de plástico reforzado con fibra de carbono especialmente ligero y robusto, si le añadimos que en su extremo el peso máximo es de 65Kgr (garra-25Kgr + Peso Saco máximo 40Kgr), se obtienen unas velocidades y aceleraciones óptimas en cada uno de los movimientos estando muy por debajo de los 180Kgr que es capaz de mover con una regulación óptima.

El sistema de agarre de sacos es la única limitación del robot en cuanto a la velocidad de paletizado. Esta mordaza permite la sujeción sin ejercer una presión sobre el mismo para no dañar el saco, pero a velocidades altas la inercia adquirida en el momento de arranque y parada puede desplazar el producto en su interior no quedando la carga homogénea en el momento de su depósito, y llegando incluso a correr el riesgo de que el saco salga disparado del sistema de agarre. Es por ello que para "**Piensos O' Couto**" no se recomienda superar una velocidad de paletizado en automático del 65%.



Como segundo elemento básico de control, el autómatas **TSX 37-10** permite una configuración libre de entradas y salidas mediante módulos y extensiones, siendo muy versátiles y adecuados para muy diferentes tipos de instalaciones. Disponen de un conector terminal serigrafiado **TER** que funciona de forma predeterminada en modalidad UNI-TELWAY maestro y permite conectar un equipo de diálogo de operador tipo XBT como el analizado en este proyecto.

Como desventaja podríamos decir que el espacio ocupado por el sistema de control es bastante voluminoso entre la CPU, los rack de extensión y módulos de entradas y salidas tipo telefax empleados, precisan de incorporar un armario eléctrico solamente para el control por autómatas. A parte nos encontramos con la necesidad de desconectar el terminal XBT para usar el PC de programación.

DESARROLLO SOFTWARE.

El software de KUKA (**Vx-Works**); a pesar de poseer una sintaxis propia, es muy familiar, pues la lógica de programación es parecida a otros lenguajes existentes, y son pocos los conocimientos necesarios para su entendimiento. Básicamente existen dos partes bien diferenciadas a la hora de desarrollar un programa en (**Vx-Works**).

1. **Programación lógica.** Se programan todas las condiciones, bucles y activación de las salidas físicas, etc.
2. **Programación de trayectorias.** Intercaladas en la parte lógica del programa, se detallan todos los puntos con sus velocidades y

	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

aceleraciones por donde ha de situarse el robot durante el programa para ejecutar cada una de las trayectorias.

El software desarrollado, permite el control automático mediante el PLC, esto dota a la instalación de un automatismo elevado, donde el operario requerirá una formación mínima ya que la instalación se manejará mediante los pulsadores de mando provistos en el armario eléctrico y el terminal de diálogo XBT, no siendo necesaria intervención alguna (salvo contadas ocasiones) en la KCP o consola del robot.

Los programas de paletización **MOSAICO_3()**, **MOSAICO_4()** y **MOSAICO_5()** así como los programas de **Zona_1 ()** y **Zona_2()** para el posicionamiento de los sacos, son fácilmente configurables, y en caso de precisar alguna configuración más de paletizado; por ejemplo **MOSIACO_6 ()** o **ZONA_3 ()**, se pueden duplicar fácilmente y editar las modificaciones posteriormente, lo que dota a este sistema de rapidez en la creación de nuevos dibujos de paletizado.

Para el autómatas, se emplea el software de programación **PL7** que ofrece 4 lenguajes IEC: lista (IL), contactos (LD), estructurado (ST) y Grafcet (SFC). Para nuestro caso se ha desarrollado un sencillo programa en lenguaje (LD) de contactos, y esto es debido a la sencillez de la periferia que integra la instalación.

Como puntos destacados, mencionamos las dos secciones con más desarrollo que integran este software, siendo estas la parte para el manejo del robot mediante automático externo y la parte de comunicaciones con el terminal XBT. Esta última parte; y como hemos venido comentando a lo largo de este proyecto, esta pensado y desarrollado para abordar y configurar un mayor número de opciones desde el terminal XBT, como pueden ser diferentes configuraciones para varias líneas, un mayor número de ordenes de robot, etc. No habiendo sido necesarias para esta instalación.

Los problemas en este sentido a nivel software, es el especial cuidado que hay que poner en la configuración de los drivers de entrada y salida del robot para la comunicación con el PLC, y el desarrollo detallado del diagrama de señales adjunto, que es valido para la puesta en marcha de los diferentes tipos de KRC que nos podemos encontrar en el mercado. Queda un poco limitado el software del PLC, pues este solamente podrá ser usado con terminales como el desarrollado en este proyecto, ya que el envío de órdenes y datos a través del terminal es muy característico para este tipo de instalaciones.

8 ANEXO.A: (Programa multitarea Sps.sub).

```

DEF SPS ( )

  LOOP
    ;Make your modifications here

;=====CONTADORES A CERO=====

if $out[10]==true then
saco_1=1
base_data[3]=base_data[1]
endif
if $out[11]==true then
saco_2=1
base_data[4]=base_data[2]
endif

;=====SIMULTANEIDAD DE LINEA=====

if (($in[13]==true) and ($in[14]==false) and ($out[30]==true)) then
$out[30]=false
endif

if (($in[13]==false) and ($in[14]==true) and ($out[30]==false)) then
$out[30]=true
endif

;=====CAPTURA DE DATOS BINARIOS=====

;;;;;;;;;;;;Capas
if (($in[15]==true) and ($in[17]==false) and ($in[18]==true) and ($in[19]==false) and
($in[20]==false)) then
reg_capas_1=registro_1
endif

;;;;;;;;;;;;Incrementos

if (($in[15]==true) and ($in[17]==false) and ($in[18]==false) and ($in[19]==false) and
($in[20]==true)) then
if $in[24]==false then
reg_alt_incr_1=registro_2
altura_1=reg_alt_incr_1*5
endif
if $in[24]==true then
reg_alt_incr_1=0-registro_2
altura_1=reg_alt_incr_1*5
endif
endif

;=====RESTOS=====



if $in[25]==true then
$out[25]=true
endif

;=====Paro temporizador=====

if $in[7]==false then
$timer_stop[1]=true
endif

    ;ENDFOLD (USER PLC)
  ENDLOOP
;FOLD ;%{H}
;FOLD
END

```


	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

9 ANEXO.B: (Programa Automático externo).

9.1 "CELL.SRC"

```

DEF CELL ( )
EXT mosaico_5 ( )
EXT mosaico_4 ( )
EXT mosaico_3 ( )
EXT home ( )

;FOLD INIT
DECL CHAR DMY[3]
DMY[]="---"
;ENDFOLD (INIT)
;FOLD BASISTECH INI
GLOBAL INTERRUPT DECL 3 WHEN $STOPMESS==TRUE DO IR_STOPM ( )
INTERRUPT ON 3
BAS (#INITMOV,0 )
;ENDFOLD (BASISTECH INI)
home ( )
;FOLD CHECK HOME
$H_POS=XHOME
IF CHECK_HOME==TRUE THEN
P00 (#CHK_HOME,#PGNO_GET,DMY[],0 ) ;Testing Home-Position
ENDIF
;ENDFOLD (CHECK HOME)
;FOLD PTP HOME Vel= 100 % DEFAULT;%{PE}%MKUKATPBASIS,%CMOVE,%VPTP,%P
1:PTP, 2:HOME, 3:, 5:100, 7:DEFAULT
$H_POS=XHOME
PDAT_ACT=PDEFAULT
BAS (#PTP_DAT )
FDAT_ACT=FHOME
BAS (#FRAMES )
BAS (#VEL_PTP,100 )
PTP XHOME
;ENDFOLD
;FOLD AUTOEXT INI
P00 (#INIT_EXT,#PGNO_GET,DMY[],0 ) ; Initialize extern mode
;ENDFOLD (AUTOEXT INI)
LOOP
P00 (#EXT_PGNO,#PGNO_GET,DMY[],0 )
SWITCH PGNO ; Select with Programnumber

CASE 1
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_5 ( ) ; Call User-Program

CASE 2
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_4 ( ) ; Call User-Program

CASE 3
P00 (#EXT_PGNO,#PGNO_ACKN,DMY[],0 ) ; Reset Progr.No.-Request
mosaico_3 ( ) ; Call User-Program

;DEFAULT
;P00 (#EXT_PGNO,#PGNO_FAULT,DMY[],0 )
ENDSWITCH
ENDLOOP
END

```

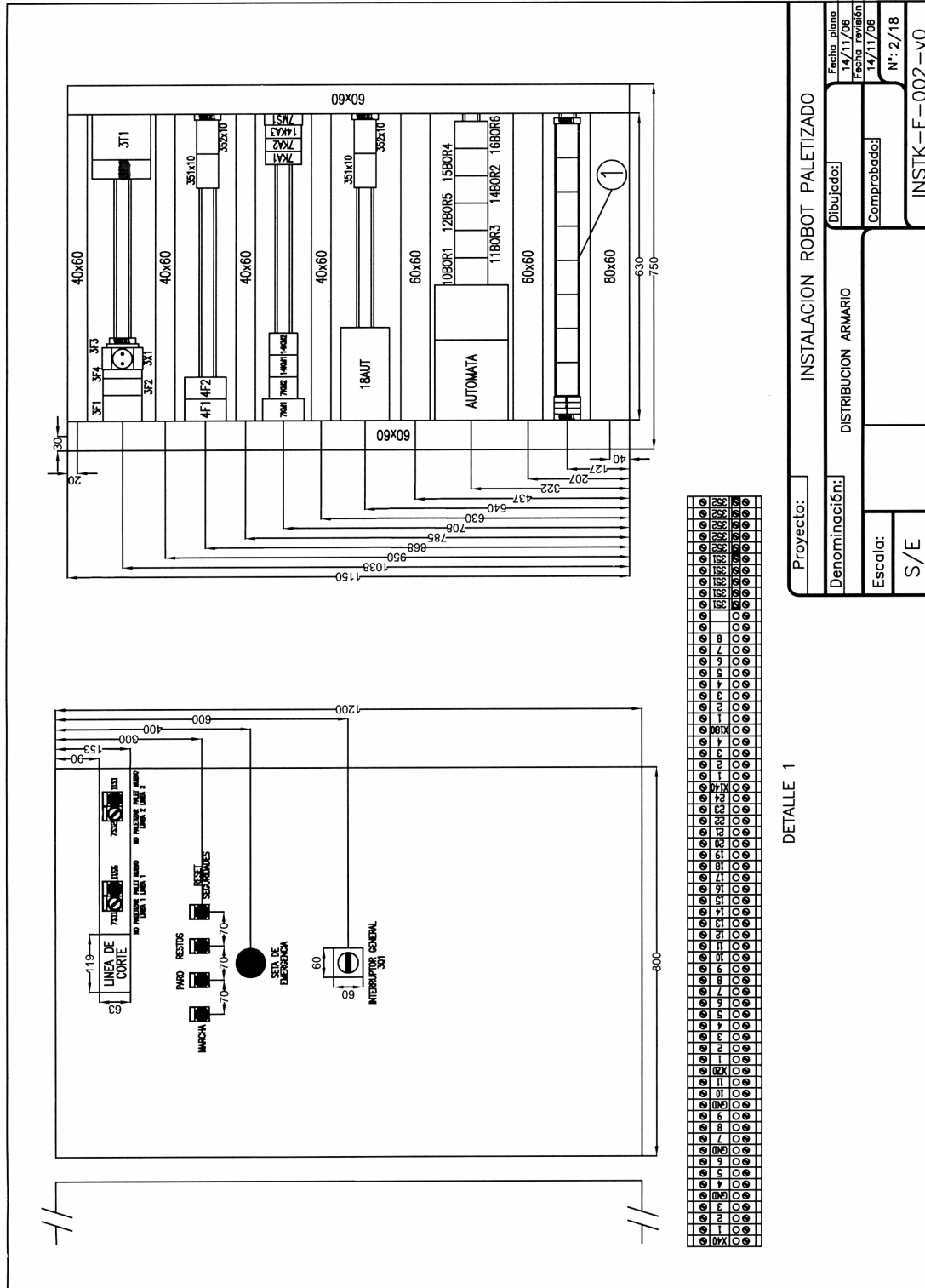
INSTALACION ROBOT PALETIZADO

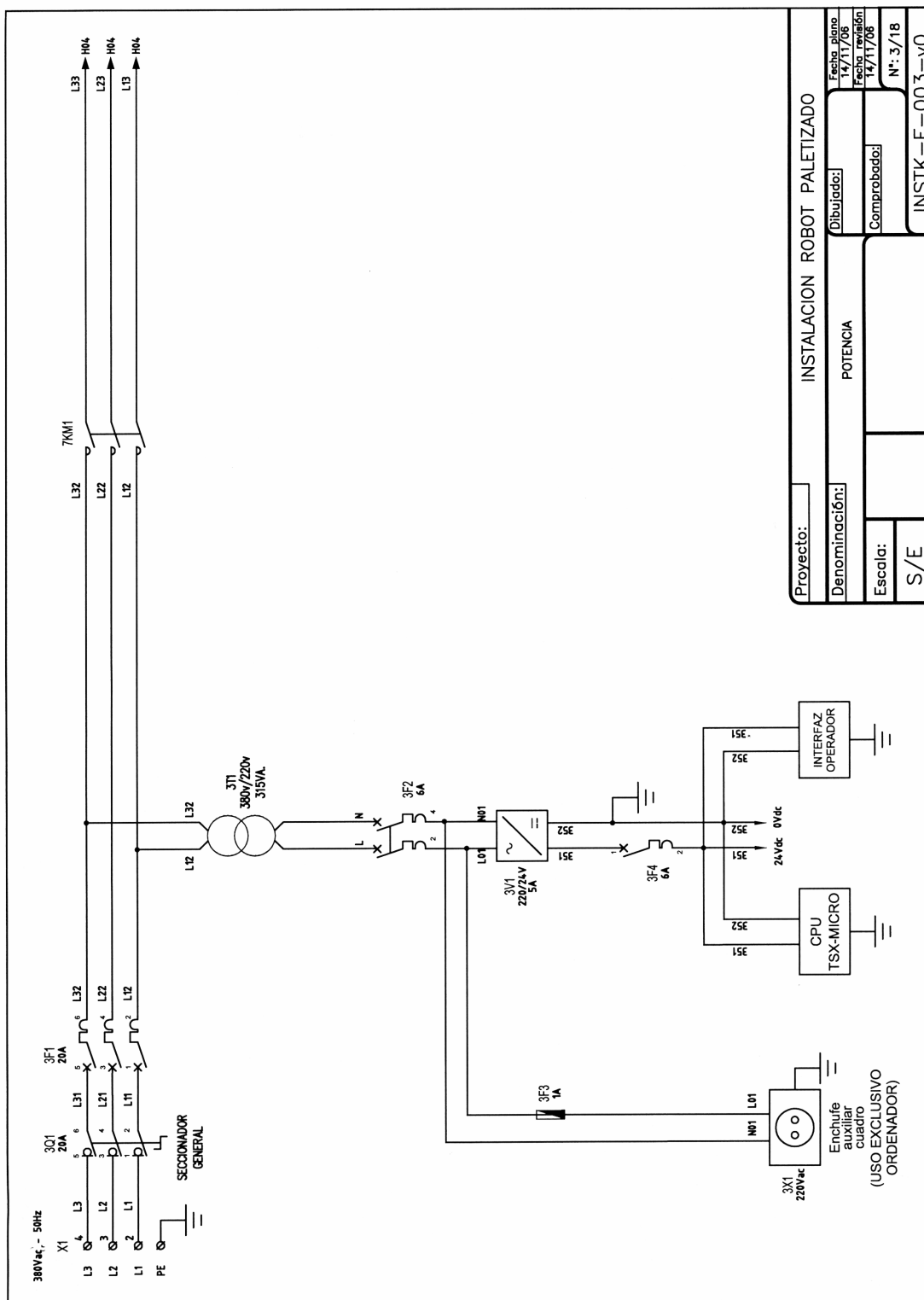
- 1- UNA LINEA DE ENTRADA Y 2-DE SALIDA.
- 2- PALETIZADO EN EL SUELO

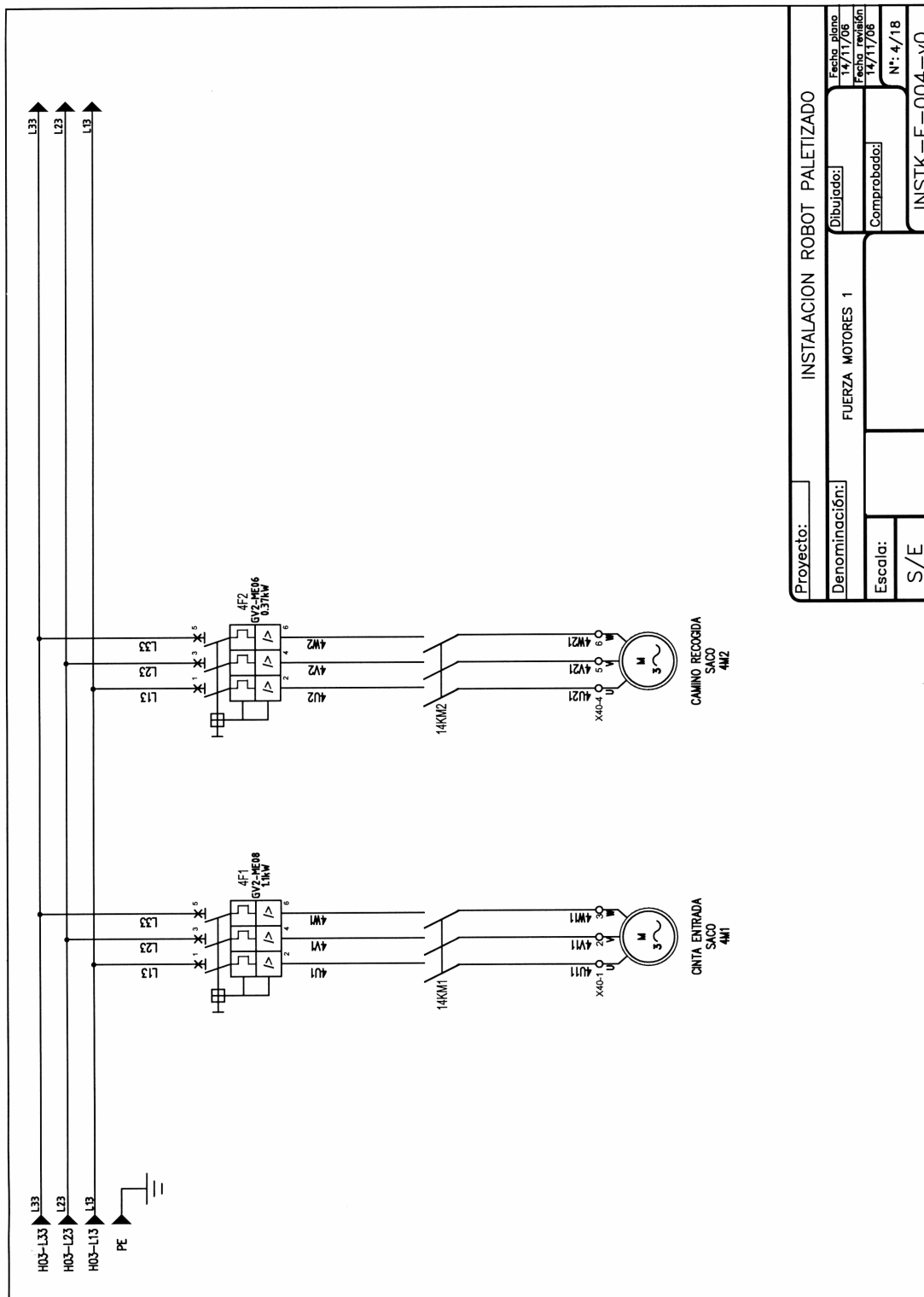
[illegible]

[illegible][illegible]

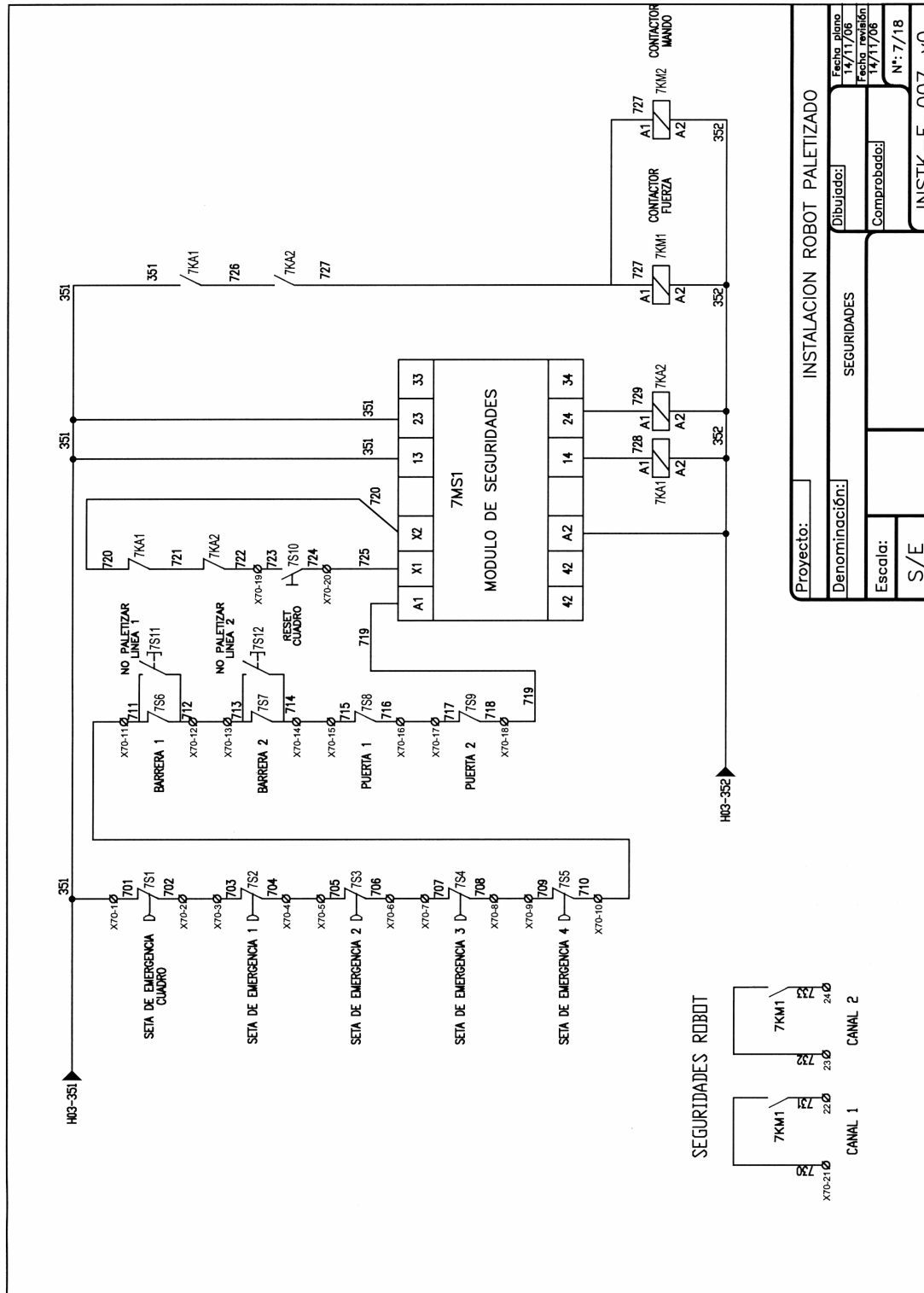
Proyecto:		INSTALACION ROBOT PALETIZADO	
Denominación:		INDICE	Fecha plano 14/11/06 Fecha revisión 14/11/06
Escala:		Comprobado:	
S/E			Nº: 1/18

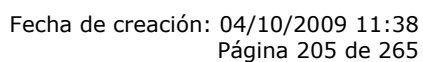


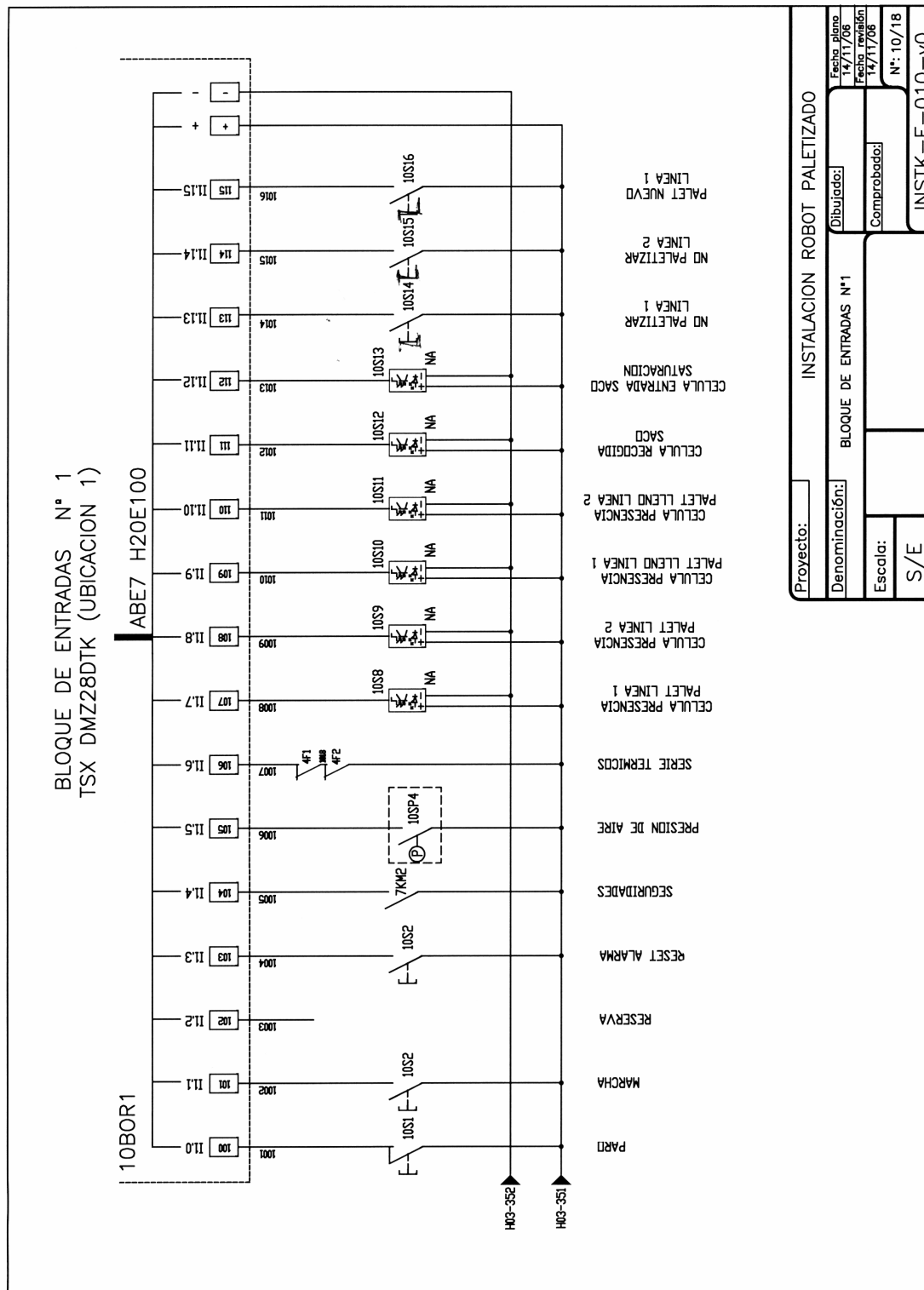


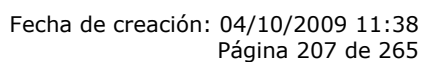


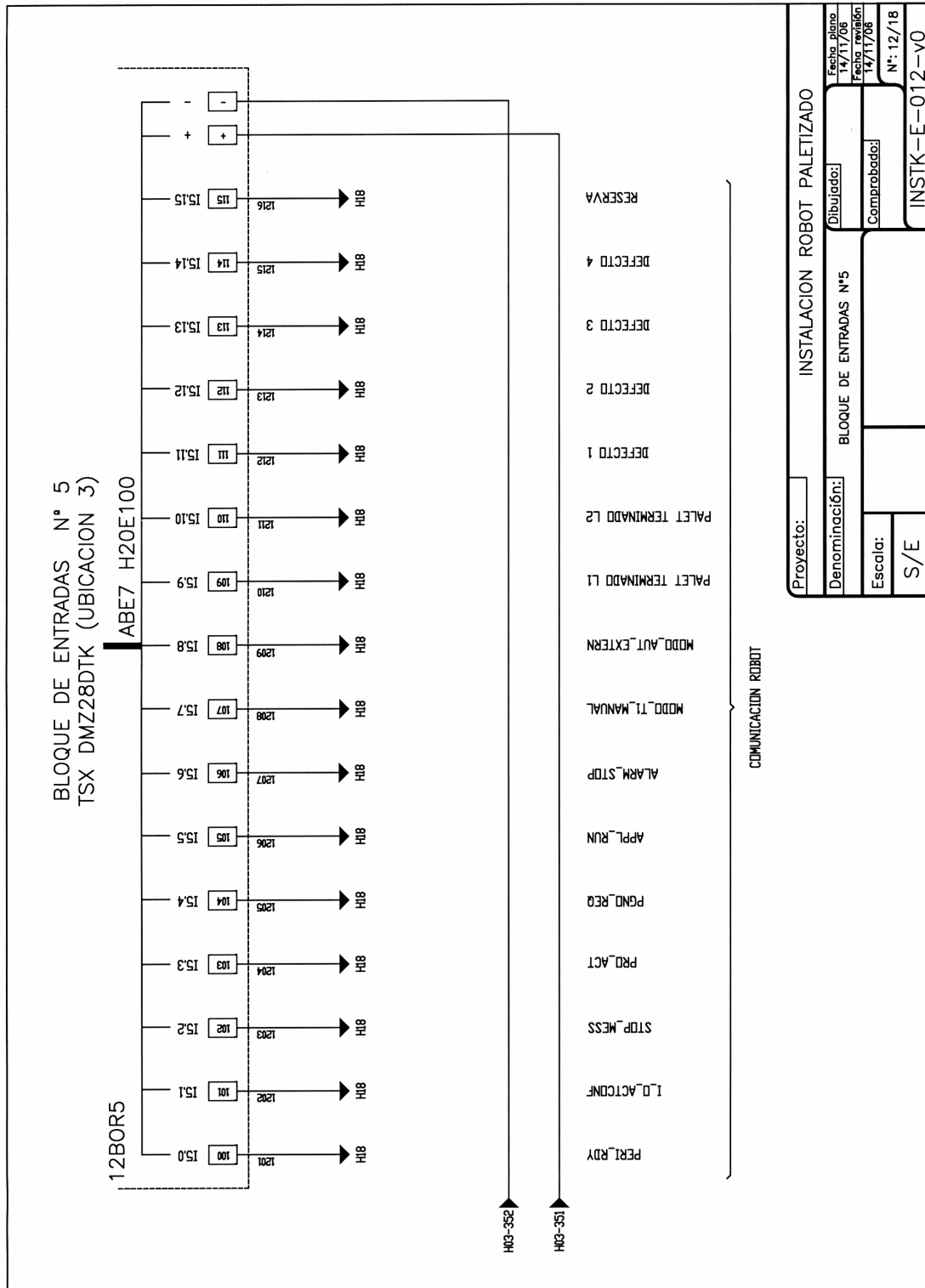
Proyecto:	INSTALACION ROBOT PALETIZADO		
Denominación:	FUERZA MOTORES 1		
Escala:	S/E		
Dibujado:	Fecha plano: 14/11/06	Fecha revisión: 14/11/06	Nº: 4/18
Comprobado:	INSTK-E-004-v0		

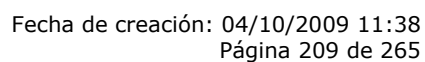


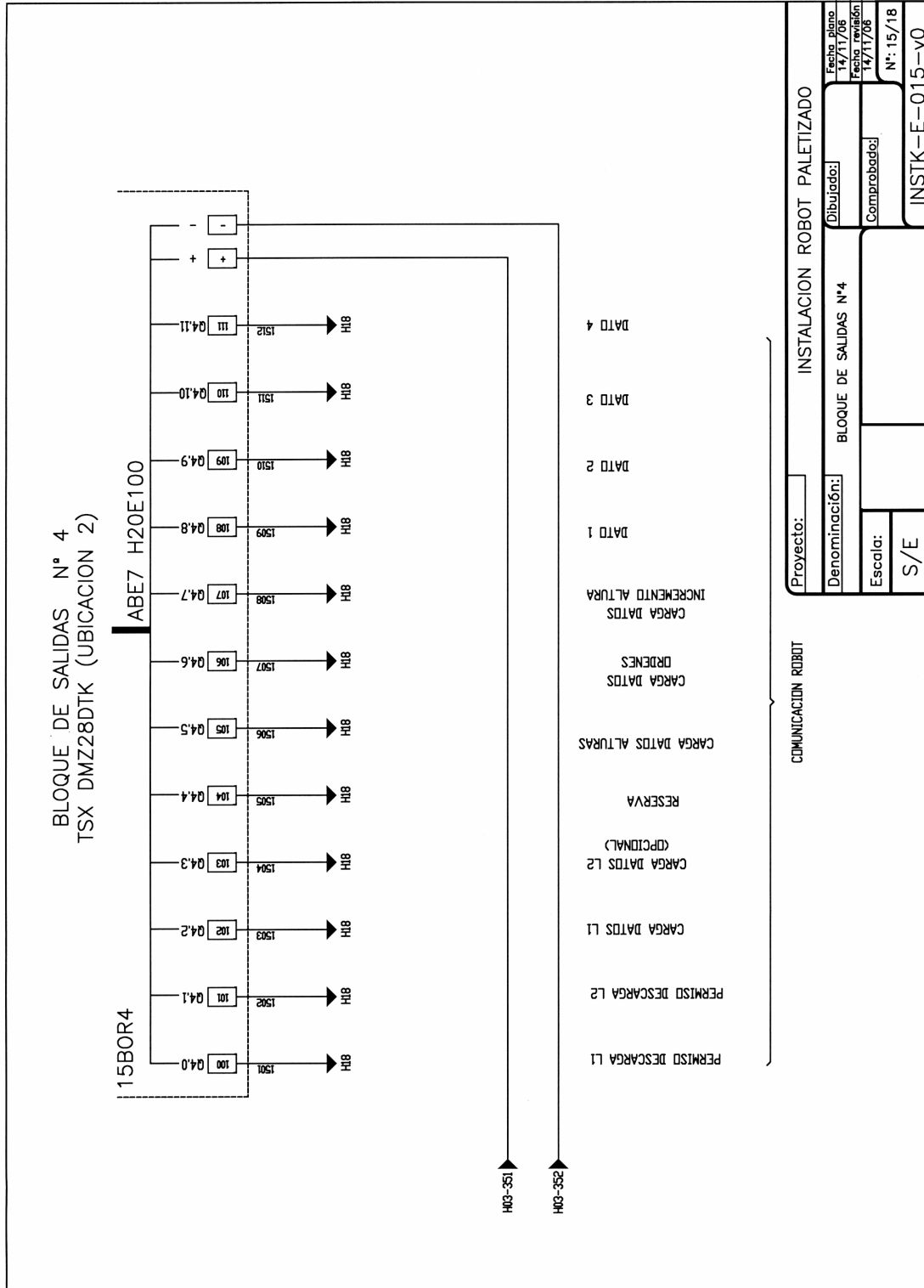


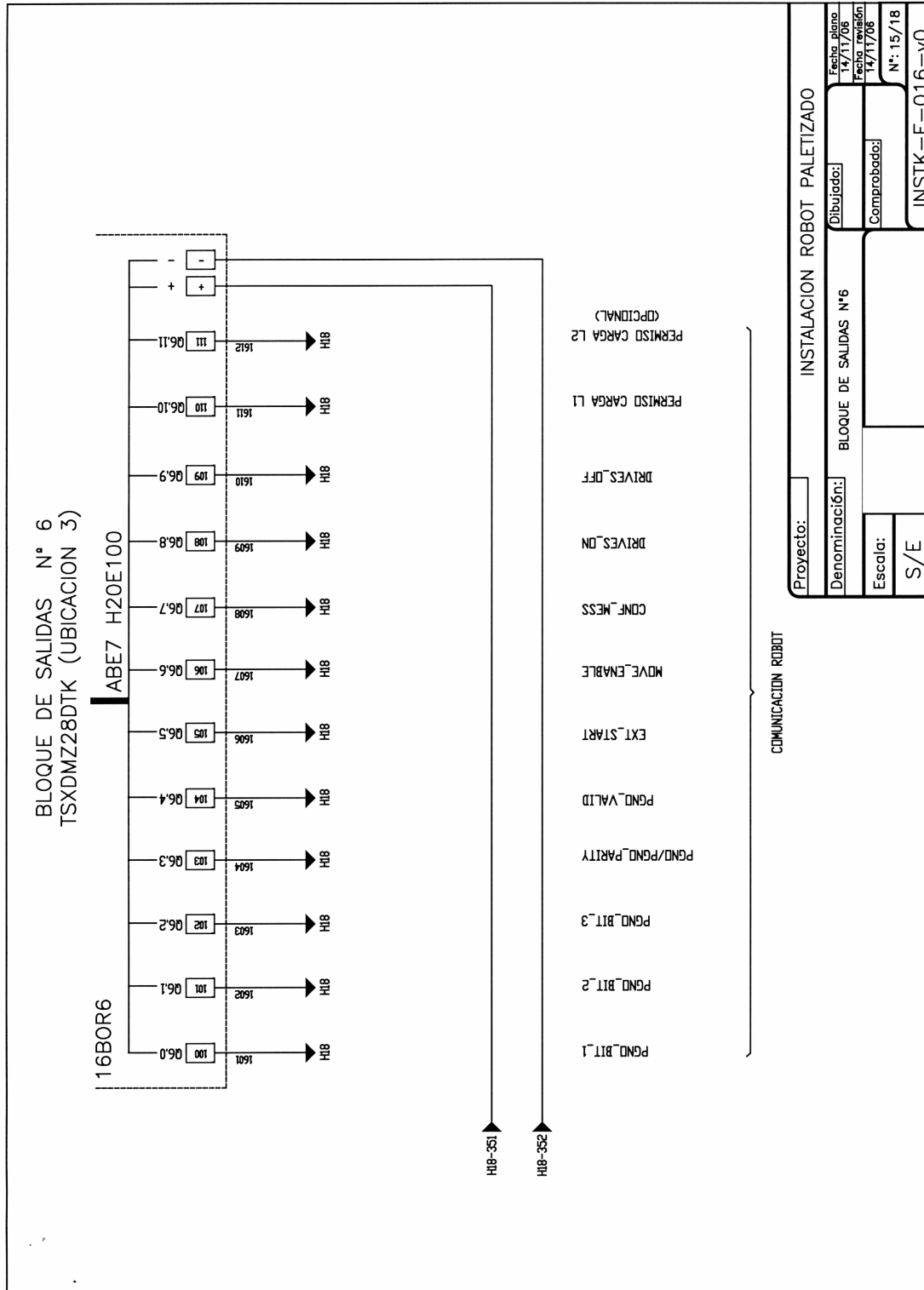


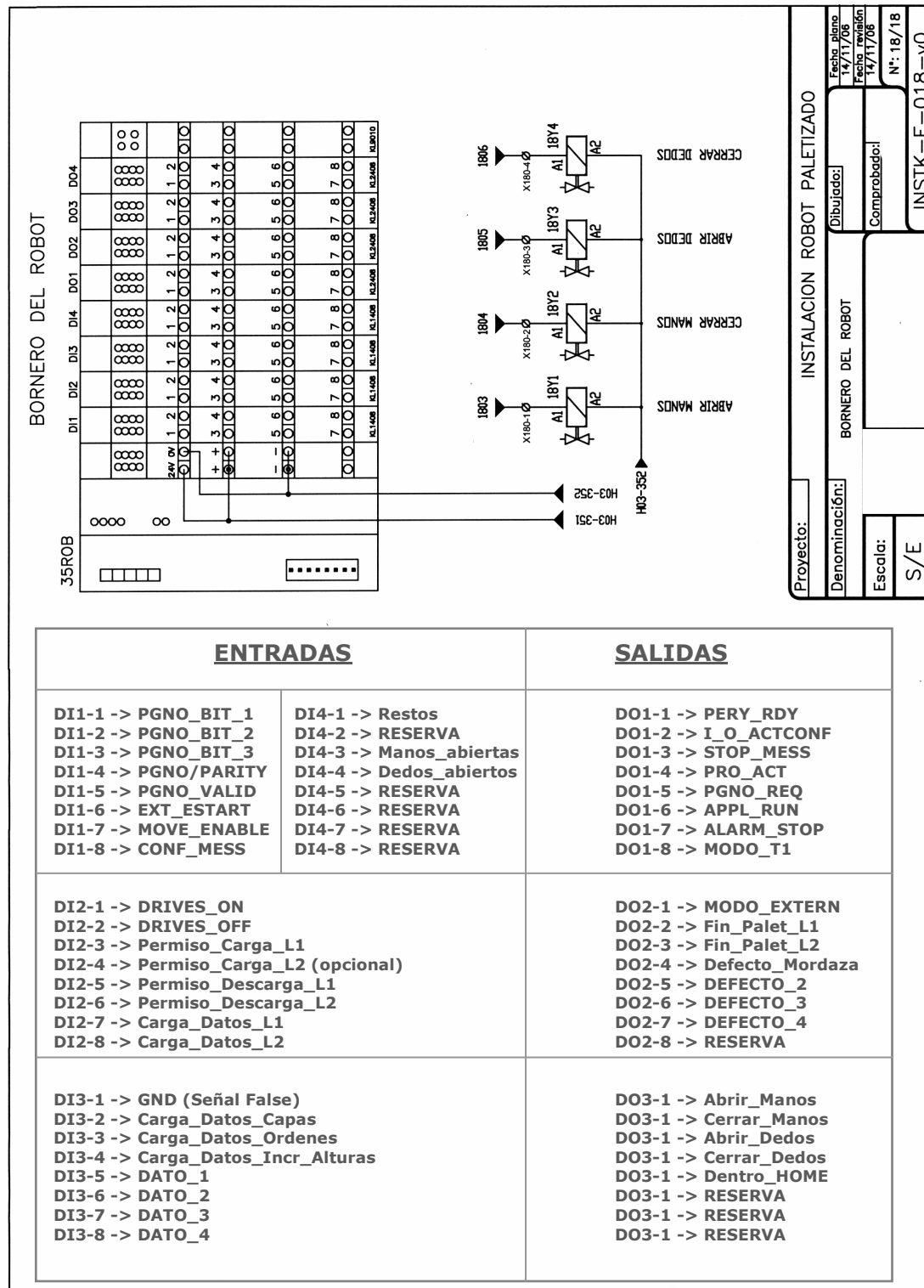














	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

11 ANEXO.D: (Programa del PLC).

11.1 VARIABLES.

11.1.1 Bit(s) Internos(s)

VARIABLE	SIMBOLO	COMENTARIO
%M0	Seguridades_activadas	
%M1	Instalacion_en_marcha	
%M4	Marcha_robot	Acciona la marcha del Robot paletizado, con el general actuando
%M5	Rearme_robot	Me proporciona los accionamientos y el reset de la interface de comunicaciones en caso de producirse una parada de emergencia
%M6	Petición_programa	Se activa con la petición del programa del KRC Pgno_req del automático externo
%M7	Sigue_peticion	
%M8	Filtrado_cel_recogida	
%M9	Filtrado_cel_rodillo	
%M10	Sigue_camino_recogida	
%M11	Se_va_a_enviar_dato_I1_a_robot	
%M12	Se_va_a_enviar_dato_I2_a_robot	
%M40	Aux_camino_recogida_saco_I1	
%M42	Aux_permiso_dejar_saco_I1	
%M43	Aux_permiso_dejar_saco_I2	
%M50	Marca_reset_contadores	

11.1.2 Palabra(s) Internas(s).

VARIABLE	SIMBOLO	COMENTARIO
%MW0	Inc_capas_por_menos_1	Incremento de capas linea multiplicado por menos 1 para los datos negativos
%MW1	Mag_mosaico	Mosaico lineas
%MW2	Mag_capas	Capas lineas
%MW3	Mag_inc_capas	Incremento de altura lineas
%MW4	Inc_capas_I2_por_menos_1	Incremento de capas linea 1 multiplicado por menos 1 para los datos negativos (RESERVA LINEA 2)
%MW5	Mag_mosaico_I2	Mosaico linea 2 (RESERVA LINEA 2)
%MW6	Mag_capas_I2	Capas linea 2 (RESERVA LINEA 2)
%MW7	Mag_inc_capas_I2	Incremento de altura linea 2 (RESERVA LINEA 2)
%MW8	Sacos_totales	Sacos totales linea
%MW9	Palet_totales	Palets totales de linea
%MW15	Mag_automatico_man	
%MW15:X0	Robot_automático_externo	
%MW18	Programa_magelis	Numero de mosaico en ejecución

%MW18:X0	Programa_magelis_bit_1	
%MW18:X1	Programa_magelis_bit_2	
%MW18:X2	Programa_magelis_bit_3	
%MW20		
%MW20:X0	Mag_cambiar_tipo_mosaico	Magelis: Cambiar tipo de mosaico linea
%MW20:X1	Mag_cambiar_capas	Magelis: Cambiar capas linea
%MW20:X2	Mag_incrementar_alturas	Magelis: Cambiar incrementar alturas linea
%MW20:X3	Mag_cambiar_tipo_mosaico_l2	Magelis: Cambiar tipo de mosaico linea 2 (RESERVA LINEA 2)
%MW20:X4	Mag_cambiar_capas_l2	Magelis: Cambiar capas linea 2 (RESERVA LINEA 2)
%MW20:X5	Mag_incrementar_alturas_l2	Magelis: Cambiar incrementar alturas linea 2 (RESERVA LINEA 2)
%MW20:X6	Mag_reset_contadores	Magelis: Ponemos a cero los contadores de producción.
%MW21		
%MW21:X0	Mag_robot_home	Magelis: Orden de robot 1
%MW21:X1	Mag_robot_mantenimiento	Magelis: Orden de robot 2
%MW21:X2	Mag_robot_liberación	Magelis: Orden de robot 3
%MW21:X3	Mag_robot_orden4	Magelis: Orden de robot 4 (RESERVA)
%MW21:X4	Mag_robot_orden5	Magelis: Orden de robot 5 (RESERVA)
%MW21:X5	Mag_robot_orden6	Magelis: Orden de robot 6 (RESERVA)
%MW21:X6	Mag_robot_orden7	Magelis: Orden de robot 7 (RESERVA)
%MW21:X7	Mag_robot_orden8	Magelis: Orden de robot 8 (RESERVA)
%MW21:X8	Mag_robot_orden9	Magelis: Orden de robot 9 (RESERVA)
%MW21:X9	Mag_robot_orden10	Magelis: Orden de robot 10 (RESERVA)
%MW21:X10	Mag_robot_orden11	Magelis: Orden de robot 11 (RESERVA)
%MW21:X11	Mag_robot_orden12	Magelis: Orden de robot 12 (RESERVA)
%MW21:X12	Mag_robot_orden13	Magelis: Orden de robot 13 (RESERVA)
%MW21:X13	Mag_robot_orden14	Magelis: Orden de robot 14 (RESERVA)
%MW22		
%MW22:X0	Mag_manual_cinta_entrada	
%MW22:X2	Mag_manual_mesa_recogida	
%MW22:X3	Mag_manual_bajar_rodillo	
%MW31	Orden_a_enviar_al_robot	
%MW118	Robot_defectos	
%MW118:X0	Robot_en_marcha	
%MW118:X1	Interface_preparada	
%MW118:X2	Accionamientos_preparados	
%MW119	Periferia_defectos	
%MW119:X0	Defecto_térmico_motores	
%MW119:X1	Defecto_seguridades	
%MW119:X2	Defecto_presión_aire	
%MW119:X3	Defecto_mordaza	

11.1.3 Temporizador(es).

VARIABLE	SIMBOLO	PRESET	MODO	TB	REG
%TM0		15	TON	10 ms	SI
%TM1	T_filtrado_bajar_rodillo	15	TON	10 ms	SI
%TM2	T_filtrado_cel_recogida	15	TON	10 ms	SI
%TM3	T_espera_dato_robot_l1	20	TON	10 ms	SI
%TM4	T_espera_dato_robot_l2	20	TON	10 ms	SI
%TM5	T_envio_dato_robot_l1	15	TON	100 ms	SI
%TM6	T_envio_dato_robot_l2	15	TON	100 ms	SI
%TM7	T_señal_mantenida_l1	10	TP	100 ms	SI
%TM8	T_señal_mantenida_l2	10	TP	100 ms	SI
%TM25	T_activar_drives_on	8	TON	100 ms	SI
%TM26	T_pulso_drives_on	2	TON	10 ms	SI
%TM27	T_conf_mess	2	TON	10 ms	SI
%TM28	T_pro_act	2	TON	10 ms	SI
%TM29	T_peticion_programa	2	TON	10 ms	SI
%TM30	T_pgno_valid	3	TON	10 ms	SI
%TM31	T_desactivacion	3	TON	10 ms	SI
%TM32	T_alarma_ioact	10	TON	1 s	SI
%TM34	T_paro_l1	15	TON	100 ms	SI
%TM63	T_camino_recogida_saco	10	TON	1 s	SI

11.1.4 Módulo de entradas_1.

VARIABLE	SIMBOLO	COMENTARIO
%I1.0	Di_boton_paro	Botón paro instalación
%I1.1	Di_boton_marcha	Botón marcha instalación
%I1.2	Reserva	
%I1.3	Di_boton_reset_alarmas	Botón reset alarmas
%I1.4	Di_seguridades	Entrada de seguridades del modulo de seguridades
%I1.5	Di_presion_aire	Entrada presion de aire de la maquina
%I1.6	Di_serie_termicos	Serie de los termicos de la instalacion
%I1.9	Di_presencia_palet_lleno_l1	Presencia palet lleno en la linea 1
%I1.10	Di_presencia_palet_lleno_l2	Presencia de palet lleno en la linea 2
%I1.11	Di_celula_recogida_saco	Celula de mesa de recogida de saco
%I1.12	Di_entrada_saco_saturacion	Celula de la cinta de entrada del saco para conseguir saturacion
%I1.13	Di_robot_no_actua_l1	El robot no paletiza en la linea de paletizado 1
%I1.14	Di_robot_no_actua_l2	El robot no paletiza en la linea de paletizado 2
%I1.15	Di_boton_palet_nuvo_l1	Palet nuevo puesto en la zona de paletizado

11.1.5 Módulo de entradas_3.



VARIABLE	SIMBOLO	COMENTARIO
%I3.0	Di_boton_palet_nuevo_I2	palet nuevo colocado en la zona de paletizado esta vacio
%I3.1	Di_presencia_rodillo	
%I3.2	Di_paro_saturacion	
%I3.3	Dr_dentro_de_home	Si estoy dentro del HOME se puede cambiar el mosaico

11.1.6 Módulo de entradas_5.

VARIABLE	SIMBOLO	COMENTARIO
%I5.0	Di_r_peri_rdy	La KRC me dice que los accionamientos estan preparados
%I5.1	Di_r_i_o_actconf	La KRC me confirma que la interface de E/S para comunicar mediante Ato.EXterno está activada.
%I5.2	Di_r_stop_mess	La KRC me informa que hay mensajes que confirmar, por paradas externas o inicio de sistema, o cualquier otro evento.
%I5.3	Di_r_pro_act	La KRC me comunica que hay un prgrama acxtivo o seleccionado, ya sea programa base cell o de paletizado.
%I5.4	Di_r_pgno_req	La KRC me solicita un numero de programa
%I5.5	Di_r_appl_run	Se está ejecutando un programa de paletizado en RUN en la KRC
%I5.6	Di_r_alarm_sto	Estando a nivel alto,el KRC me informa de haberse producido un evento de parada de emergencia (nivel bajo).
%I5.7	Di_r_mod0_t1_manual	El KRC me indica mediante esta señal, que la llave se encuentra en el modo de operación MANUAL.
%I5.8	Di_r_mod0_aut_externo	El KRC me indica mediante esta señal, que la llave se encuentra en el modo de operación Automático_Externo.
%I5.9	Di_palet_terminado_I1	el paletizado se ha terminado en la linea 1
%I5.10	Di_palet_terminado_I2	el paletizado se ha terminado en la linea 2
%I5.11	Di_defecto_mordaza	

11.1.7 Módulo de salidas_2.

VARIABLE	SIMBOLO	COMENTARIO
%Q2.0	Do_cinta_entrada_saco_I1	cinta de entrada de saco de la linea 1
%Q2.1	Do_camino_recogida_saco_I1	camino de recogida de saco de linea 1
%Q2.2	Do_paro_I1	Paro línea 1
%Q2.3	Do_robot_en_marcha	Robot en marcha
%Q2.4	Do_bajar_rodillo	Rodillo para prensar el saco



	Proyecto	
	Documento	
	PROYECTO FIN DE CARRERA	
	Instalación de Robot Paletizado.	

11.1.8 Módulo de salidas_4.

VARIABLE	SIMBOLO	COMENTARIO
%Q4.0	Do_permiso_descarga_l1	
%Q4.1	Do_permiso_descarga_l2	
%Q4.2	Do_robot_carga_datos	ROBOT: Carga datos linea
%Q4.3	Do_robot_carga_datos_l2	ROBOT: Carga datos linea 2 (OPCIONAL)
%Q4.4	Do_robot_carga_datos_mosaico	ROBOT: Carga datos de mosaico
%Q4.5	Do_robot_carga_datos_capas	ROBOT: Carga datos de numero de capas
%Q4.6	Do_robot_carga_datos_ordenes	ROBOT: Carga datos de ordenes
%Q4.7	Do_robot_carga_datos_incremento	ROBOT: Carga datos de incremento de alturas
%Q4.8	Do_robot_dato1	ROBOT: Dato 1
%Q4.9	Do_robot_dato2	ROBOT: Dato 2
%Q4.10	Do_robot_dato3	ROBOT: Dato 3
%Q4.11	Do_robot_dato4	ROBOT: Dato 4

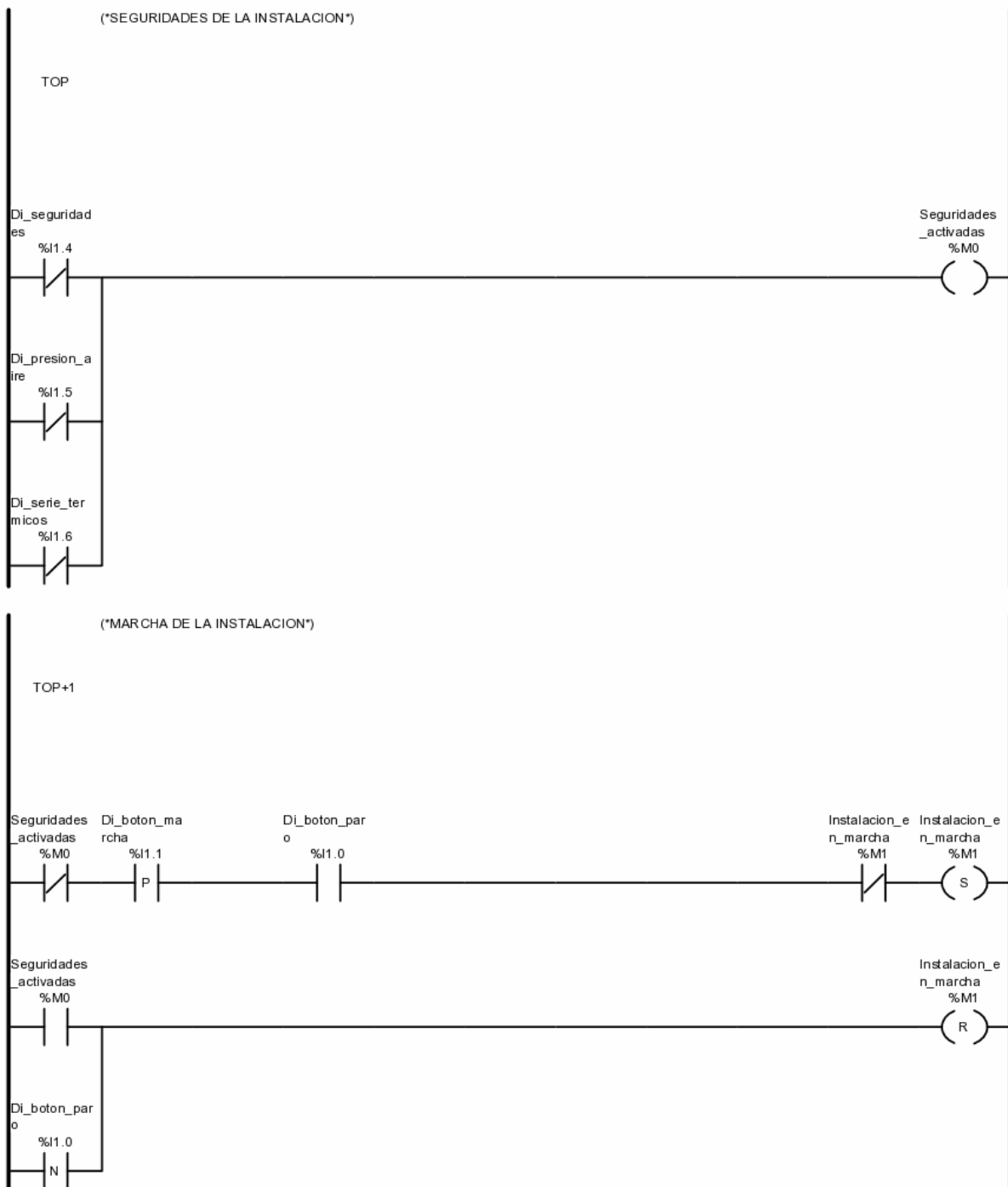
11.1.9 Módulo de salidas_6.

VARIABLE	SIMBOLO	COMENTARIO
%Q6.0	Do_r_pgno_bit_1	Bit 1 del código binario para determinar el programa
%Q6.1	Do_r_pgno_bit_2	Bit 2 del código binario para determinar el programa
%Q6.2	Do_r_pgno_bit_3	Bit 3 del código binario para determinar el programa
%Q6.3	Do_r_pgno_pgno_parity	Bit que representa la preasignación al envío de programa y la paridad en caso de estar habilitada
%Q6.4	Do_r_pgno_valid	Decimos al KRC que hemos transmitido el nº de programa, validando así el dato
%Q6.5	Do_r_ext_start	Señal para iniciar o continuar un programa en la interface E/S del KRC
%Q6.6	Do_r_move_enable	Habilitamos el movimiento del KRC, pudiendo realizarse desplazamientos manuales, ejecutar o parar un programa.
%Q6.7	Do_r_conf_mess	Confirmamos al KRC todos los tipos de mensajes confirmables una vez subsanada la causa que los produjo.
%Q6.8	Do_r_drives_on	Activamos accionamientos del KRC
%Q6.9	Do_r_drives_off	Desactivamos accionamientos del KRC
%Q6.10	Do_r_permiso_carga	

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

11.2 PROGRAMACIÓN EN CONTACTOS.

11.2.1 LD_Mast_General.



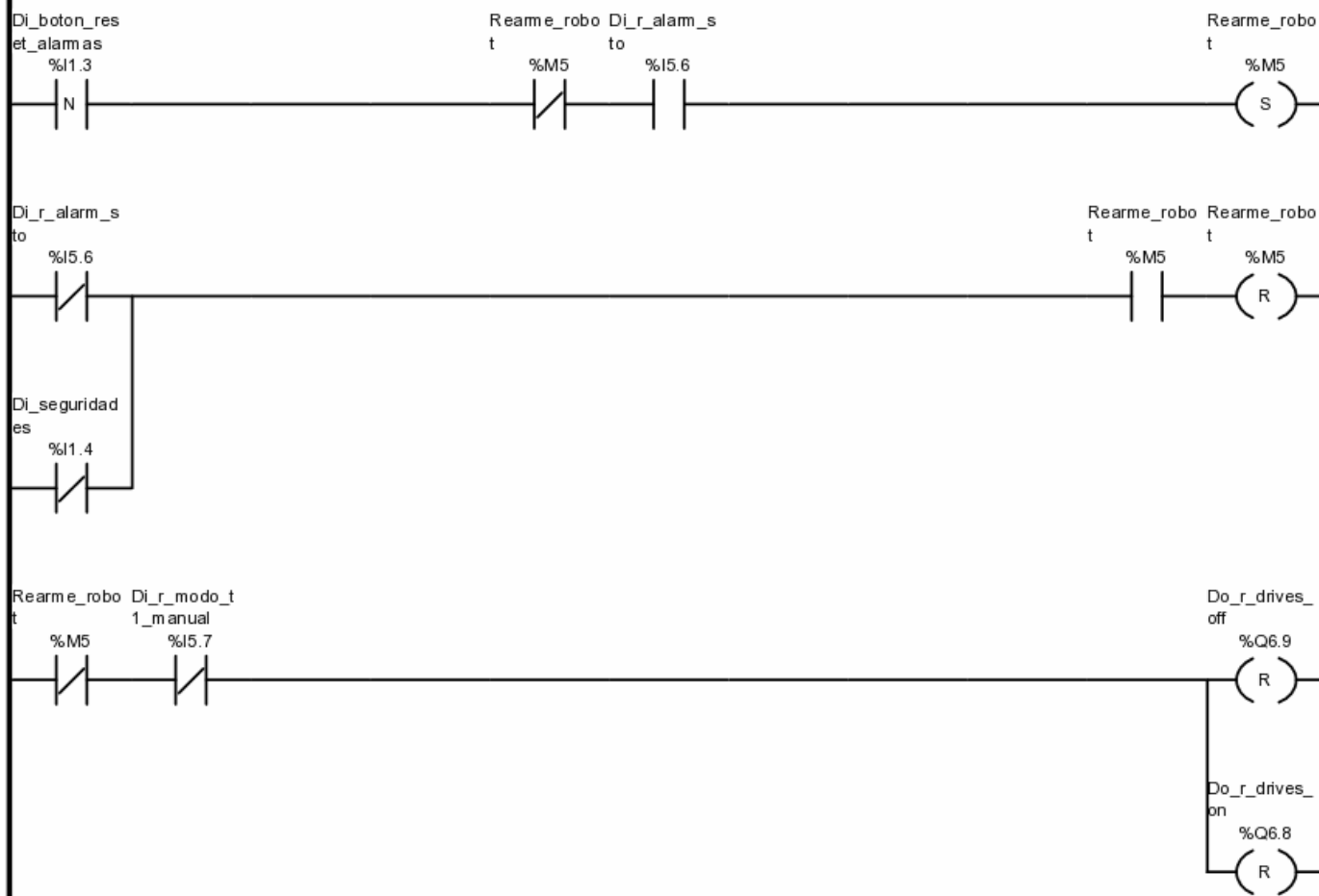
Carpeta: 1

MAST - General

Fecha: 28/8/09

(*Rearme de la Instalacion Robot. Para el marcha del Robot y activacion de los Accionamientos del Robot drives_off y Drves_on*)

TOP+1



Carpeta: 2

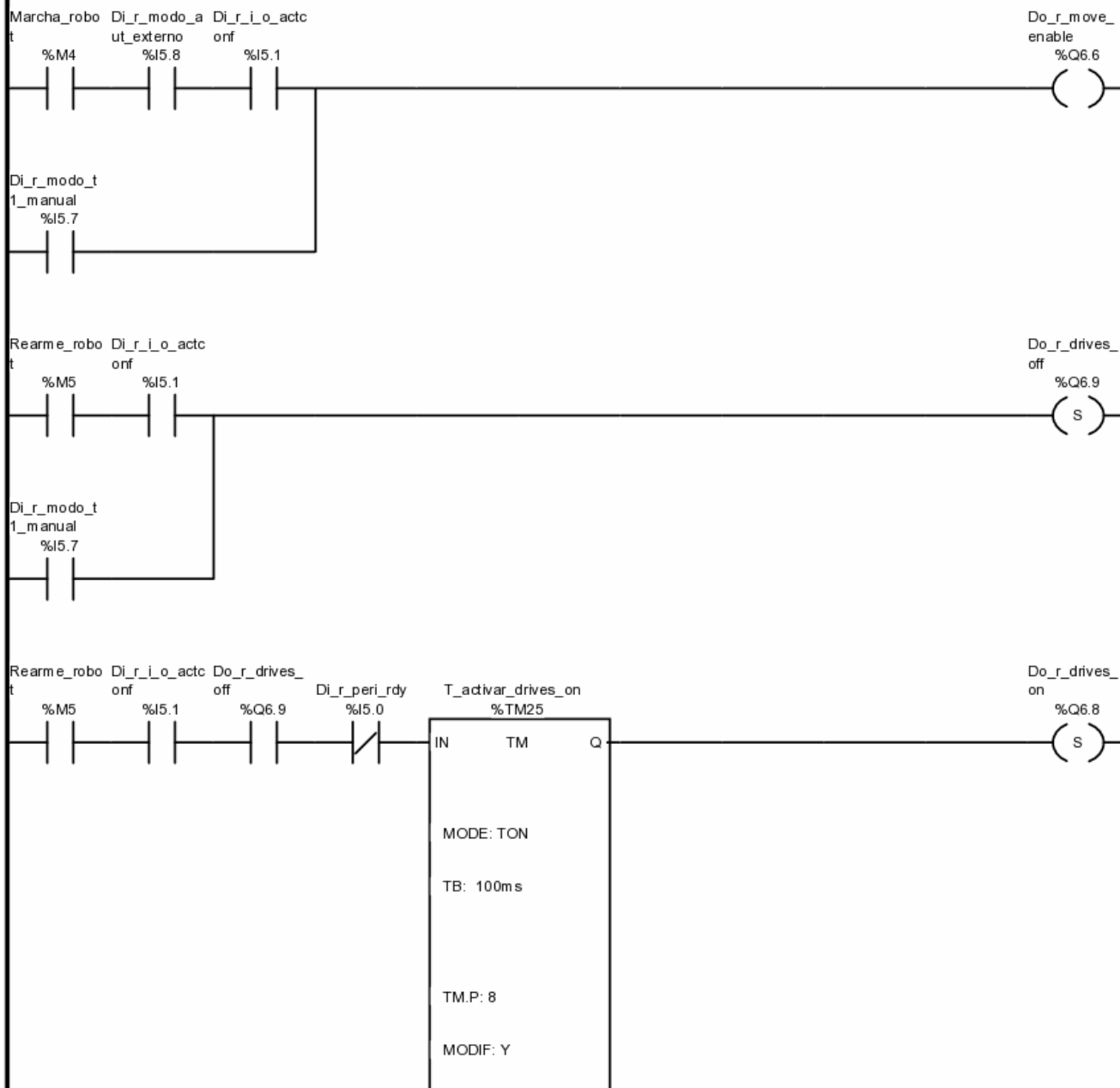
MAST - Robot_auto_exter

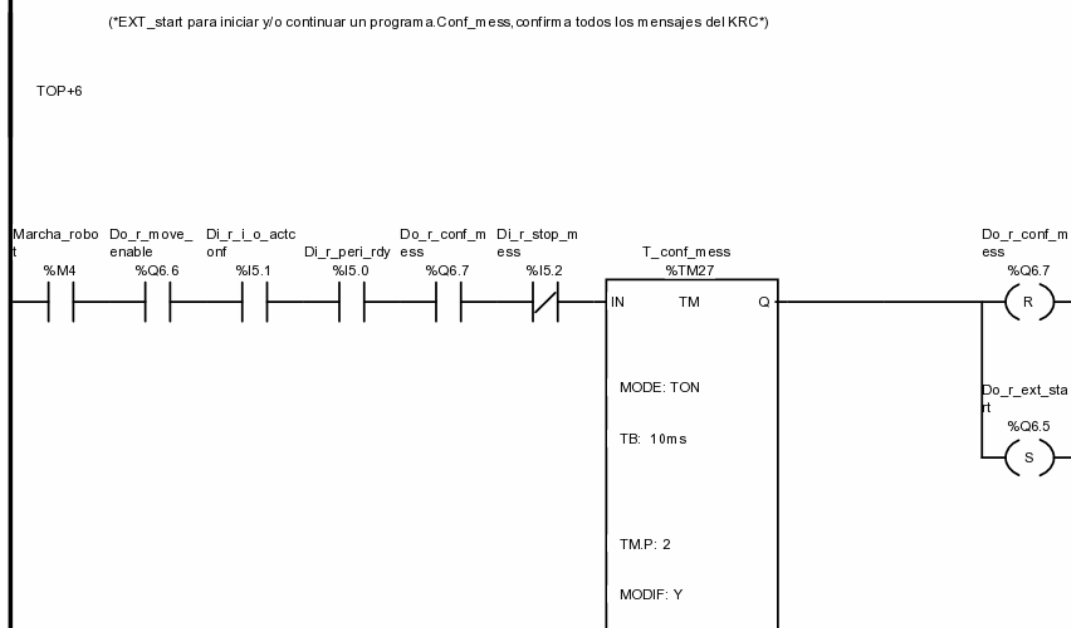
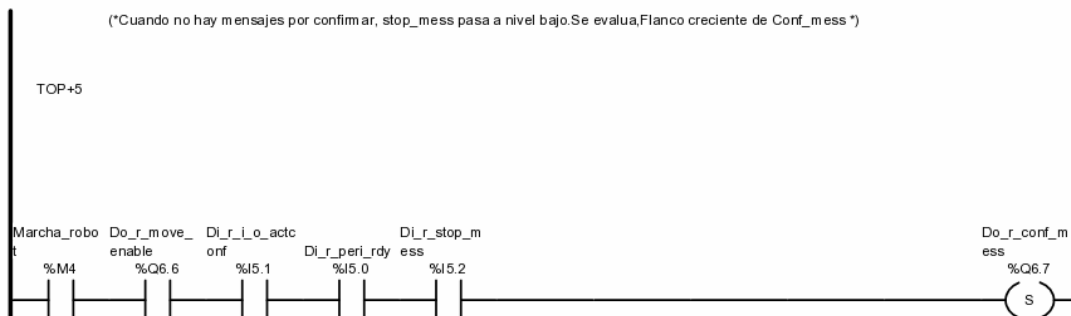
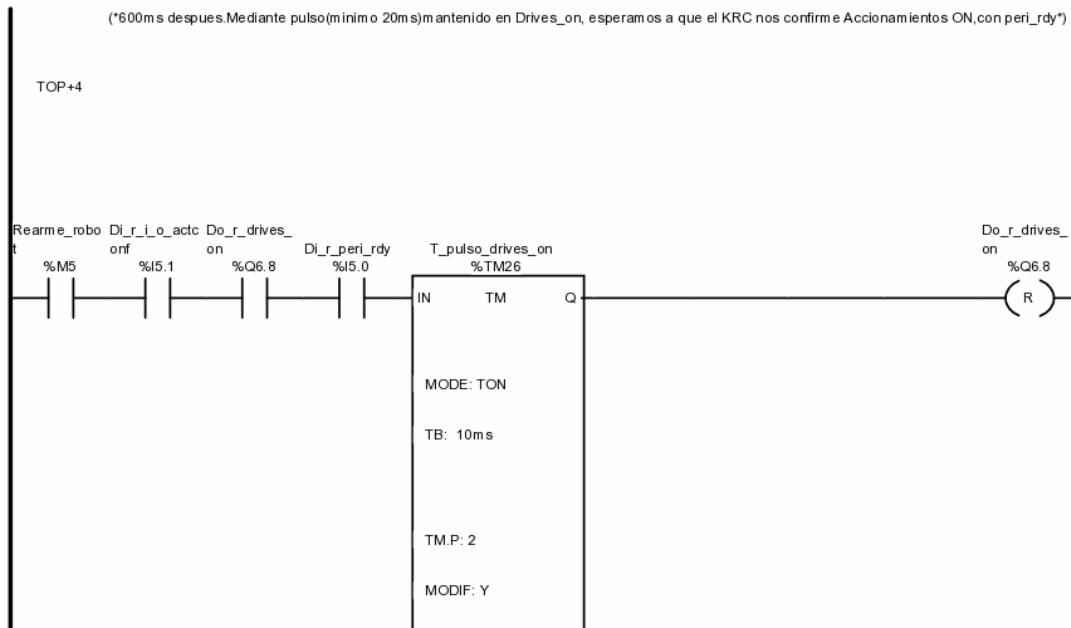
Fecha: 28/8/09

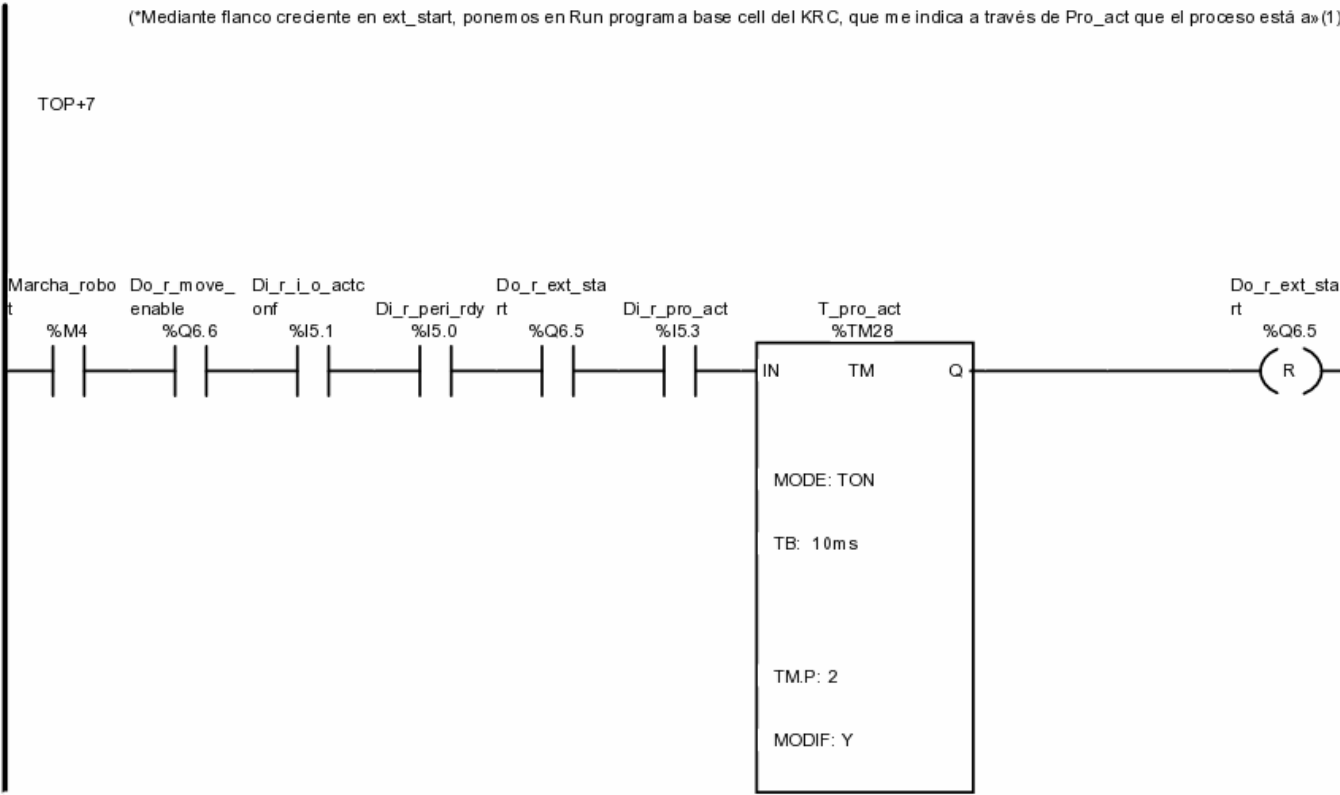


(*Con la interface i_o_act activada y en modo Aut.Externo.Habilitamos señal Move_enable para el movimiento del KRC, y damos Accionamientos*)

TOP+3





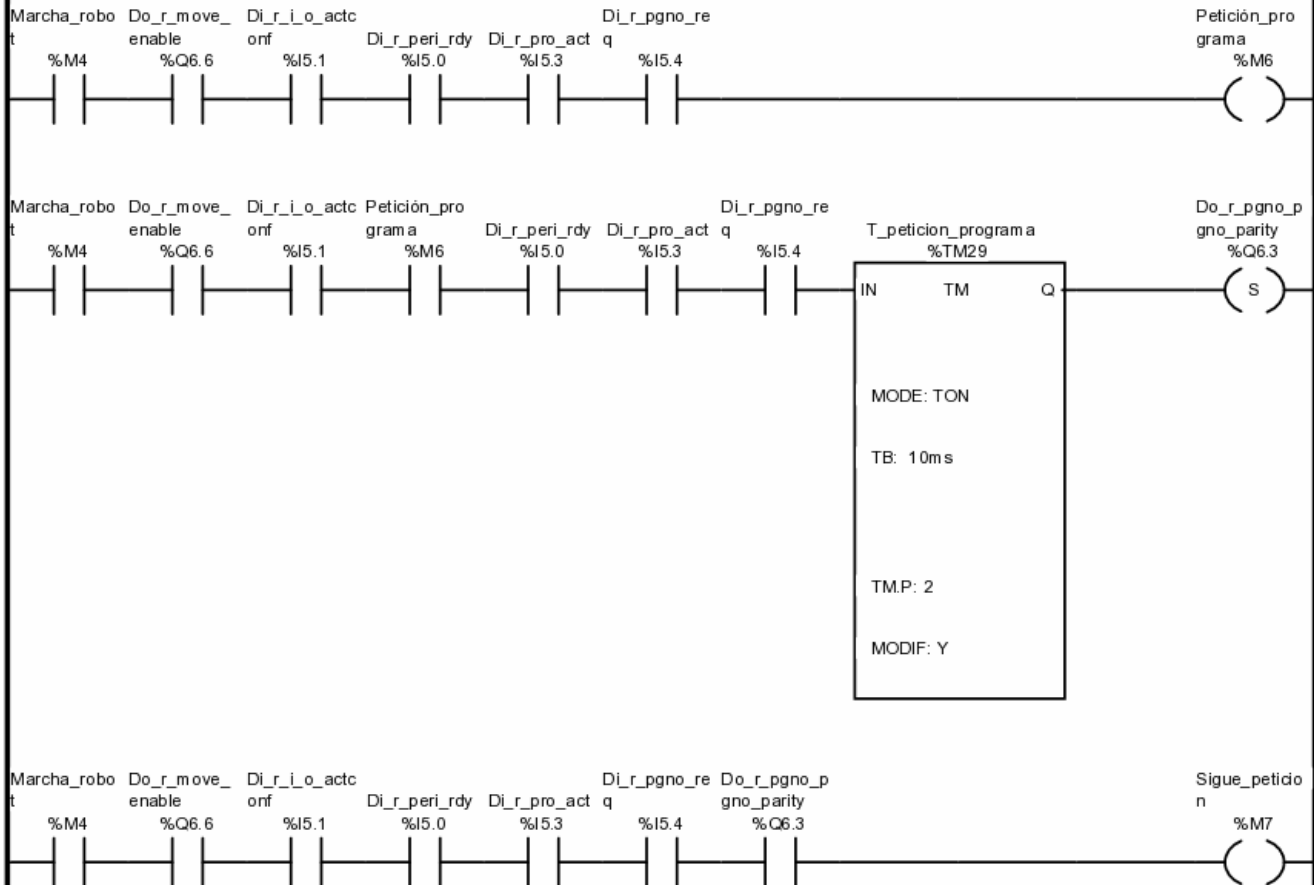


Lista de los reenvios del comentario:

(1):(*Mediante flanco creciente en ext_start, ponemos en Run program a base cell del KRC, que me indica a través de Pro_act que el proceso está activo*)

(*Con el programa base Cell.Src activo, se nos genera la petición Pgno_req para la emisión del número de programa, y solidud de Pgno_Parity*)

TOP+8

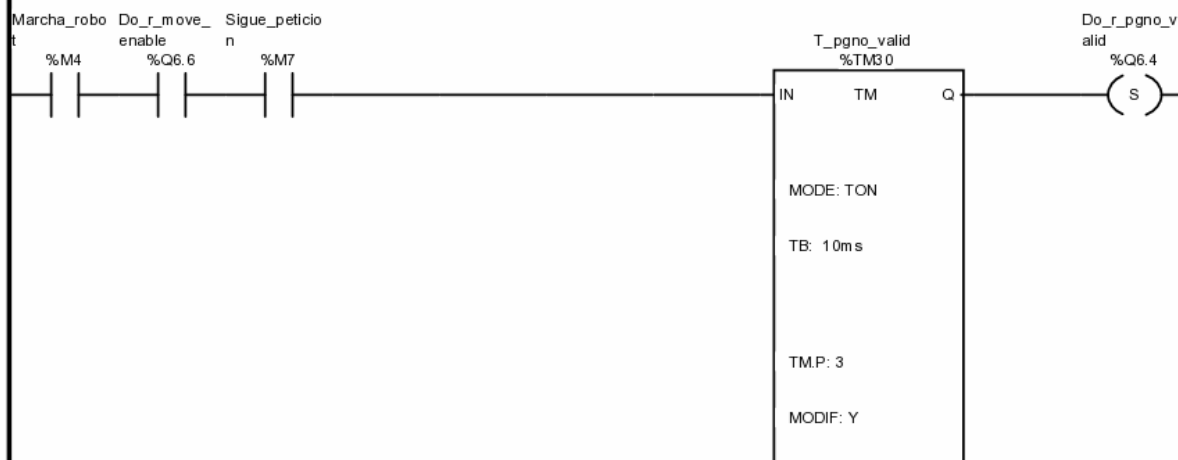


Carpeta: 7

MAST - Robot_auto_exter

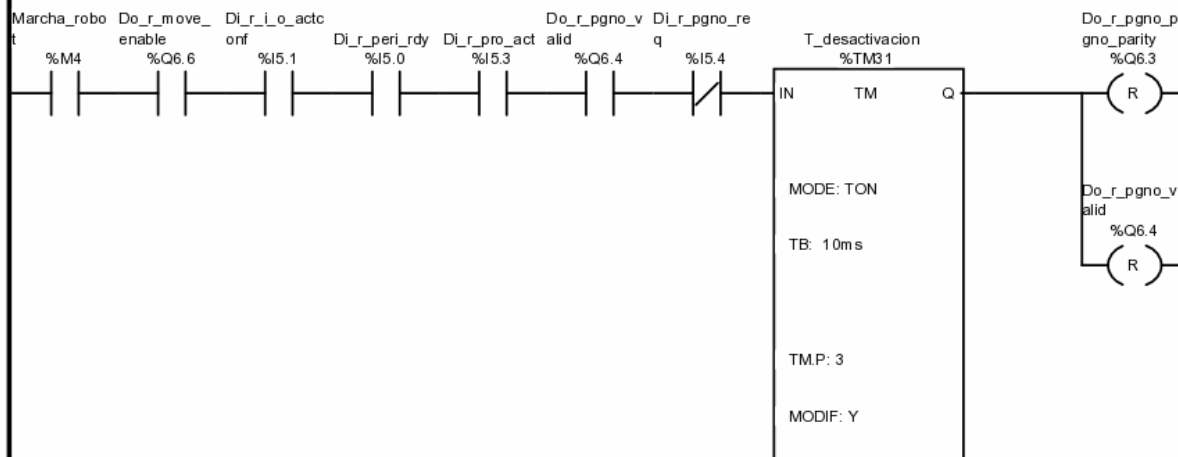
Fecha: 28/8/09


TOP+9



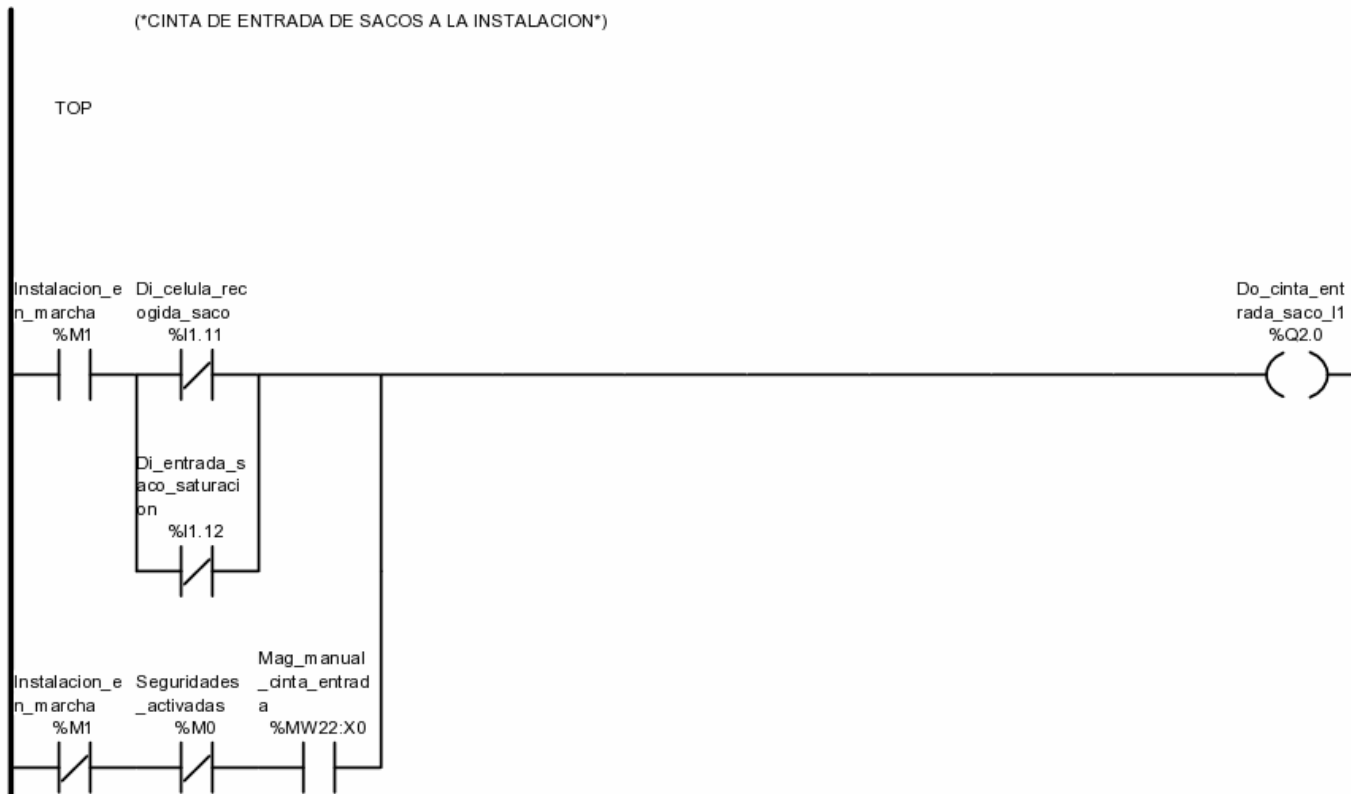
("Cuando el KRC, admite el tipo de programa, Pgno_req pasa a estado bajo y me acciona el RUN del programa mediante Appl_run")

TOP+10



ROBITEK.System Dpto.Robótica	<i>Proyecto</i> PROYECTO FIN DE CARRERA	
	<i>Documento</i> Instalación de Robot Paletizado.	

11.2.3 LD_Mast_Recogida_De_Saco.



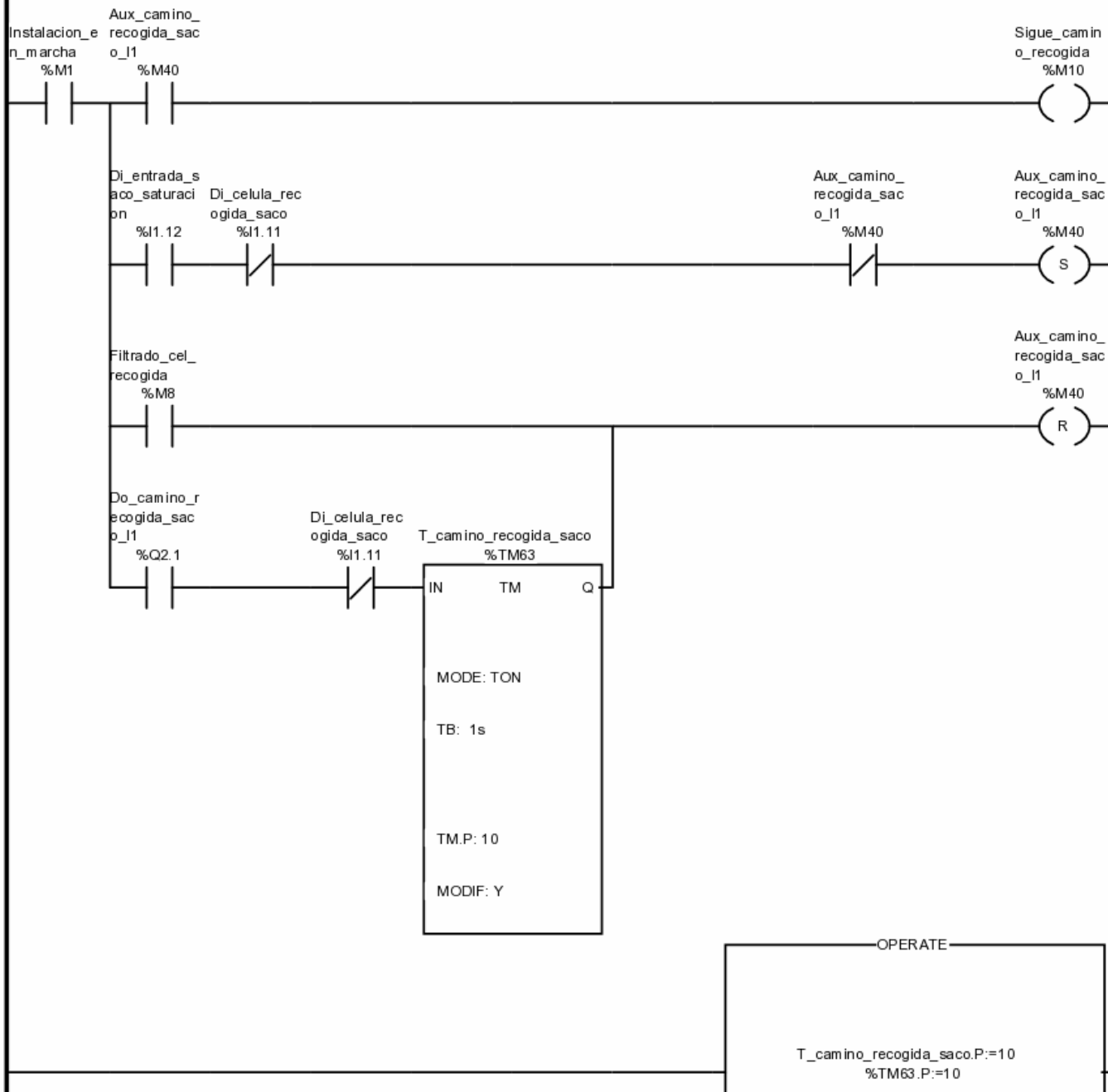
Carpeta: 1

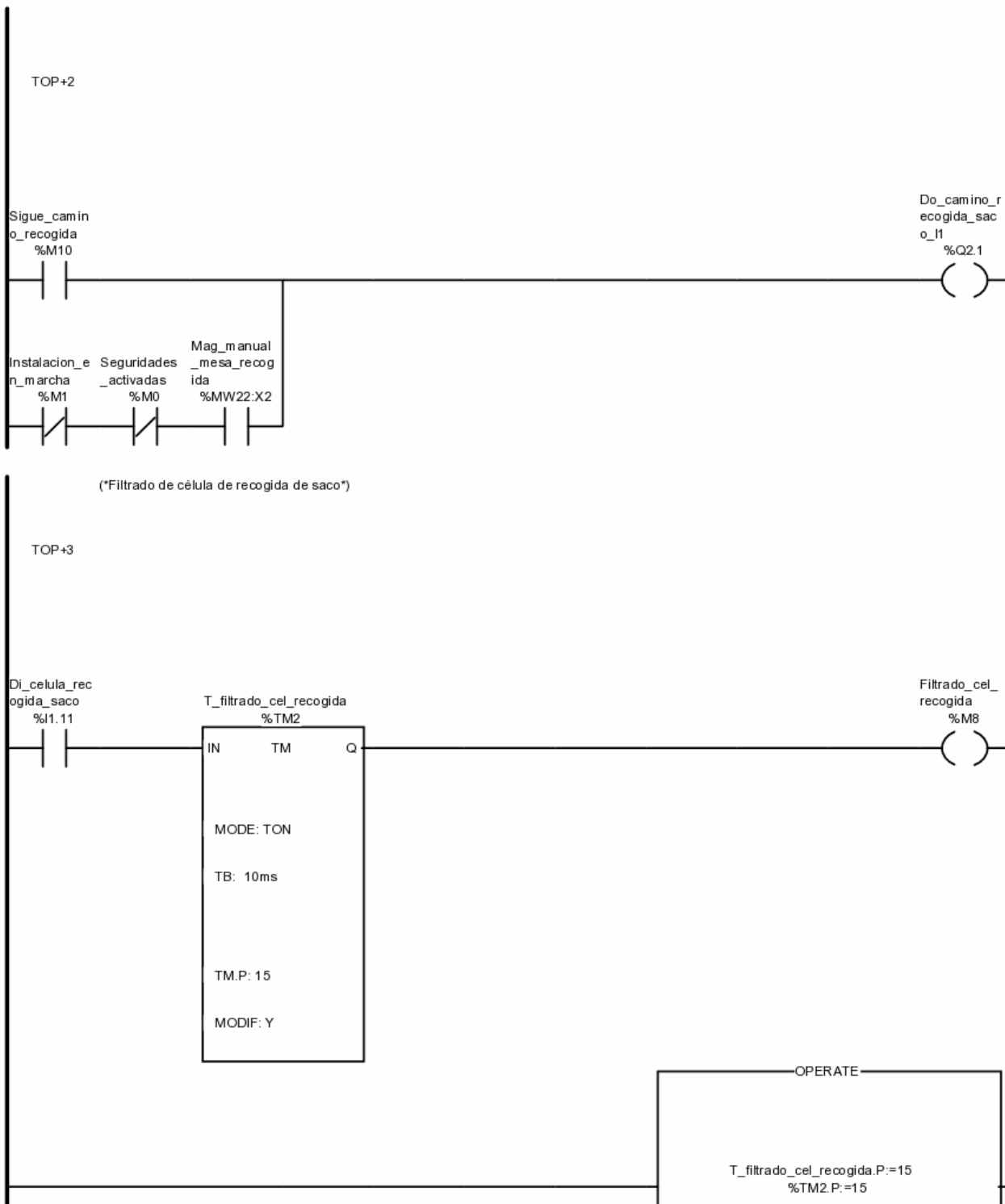
MAST - Recogida_de_saco

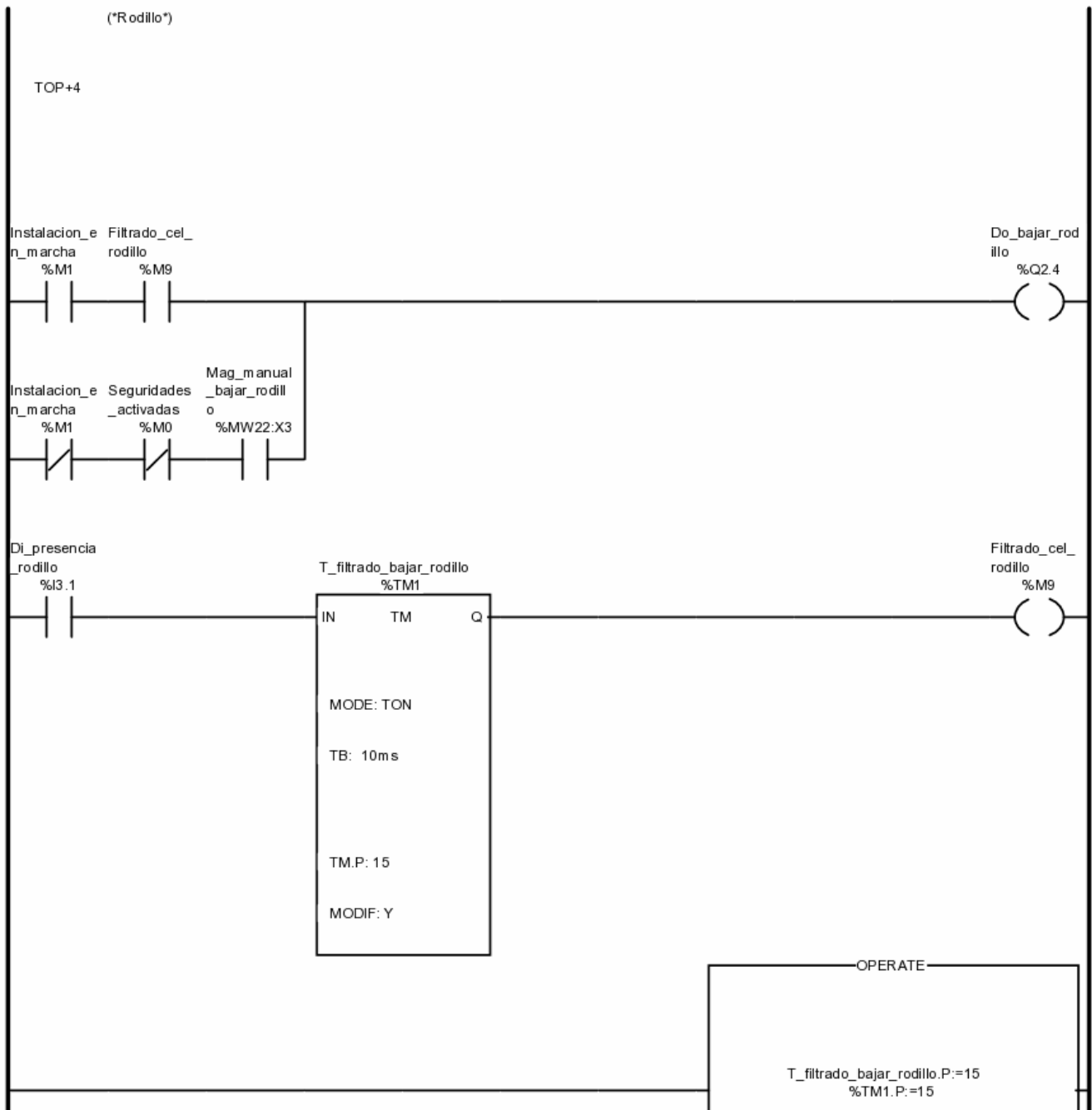
Fecha: 28/8/09

(*CAMINO DE RECOGIDA DE SACO DE LA INSTALACION*)

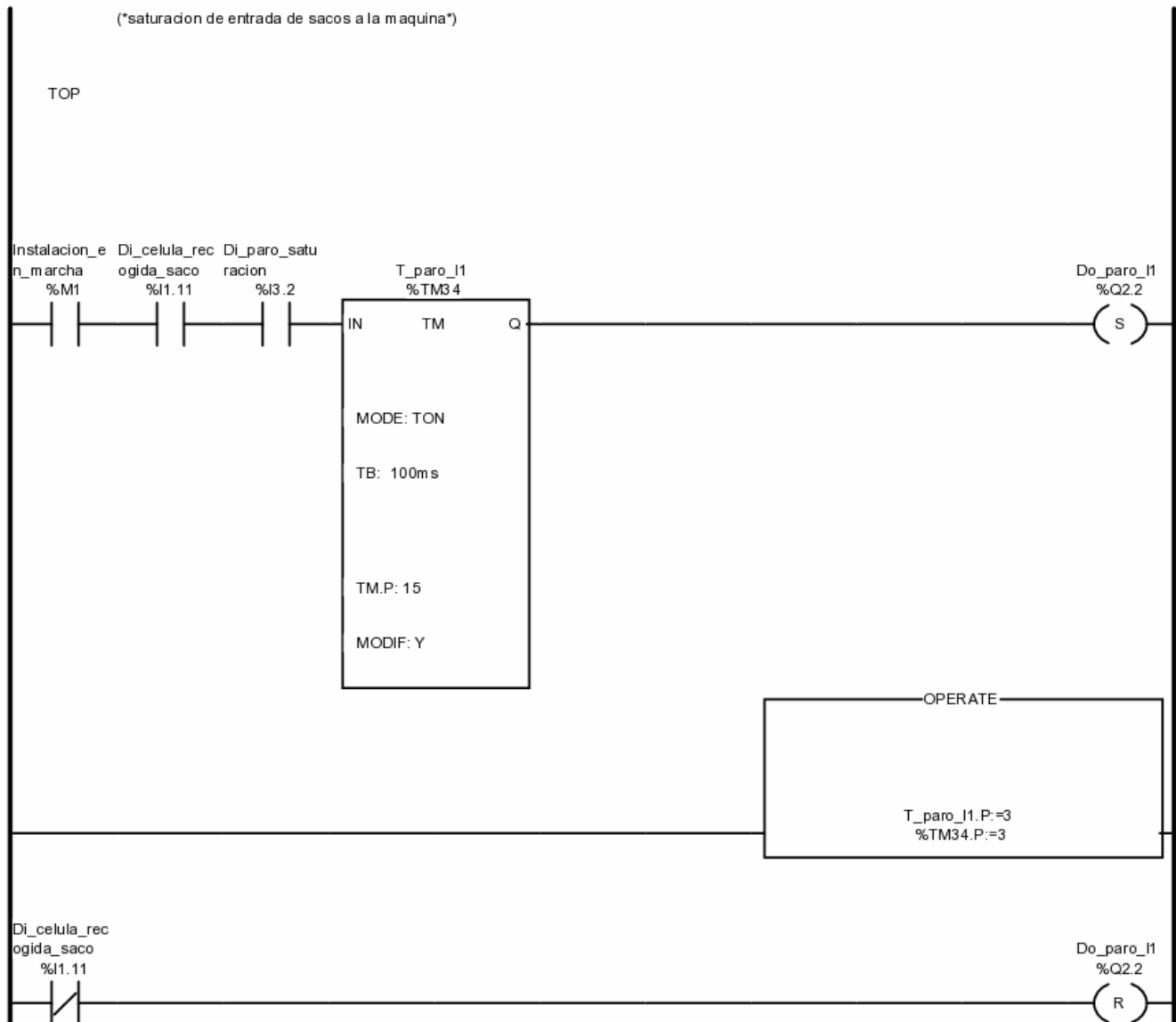
TOP+1







11.2.4 LD_Mast_Saturación.

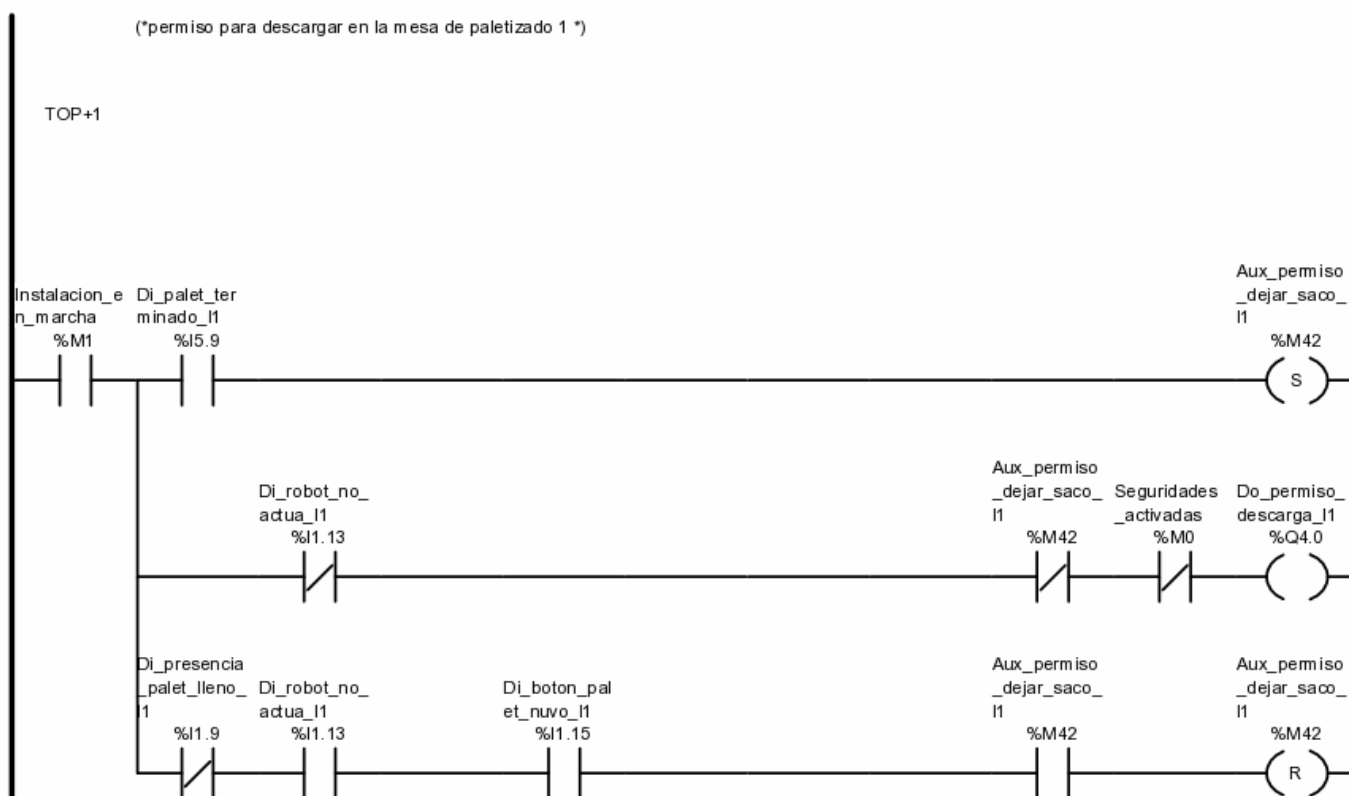
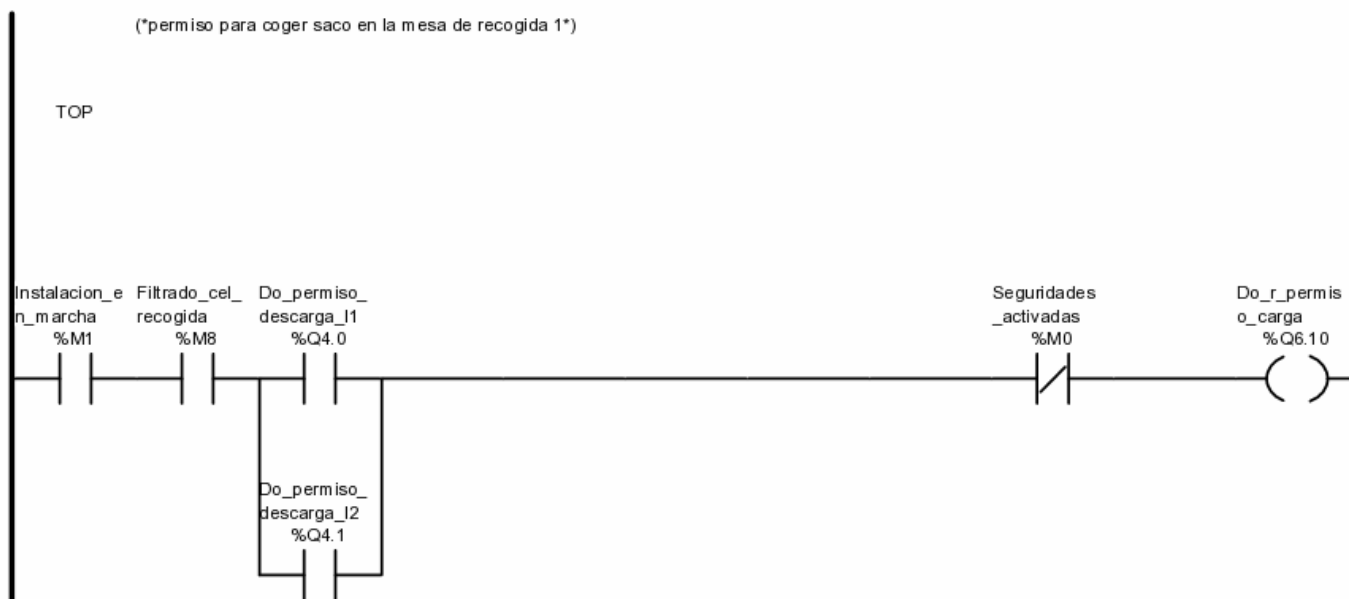


Carpeta: 1

MAST - Saturacion

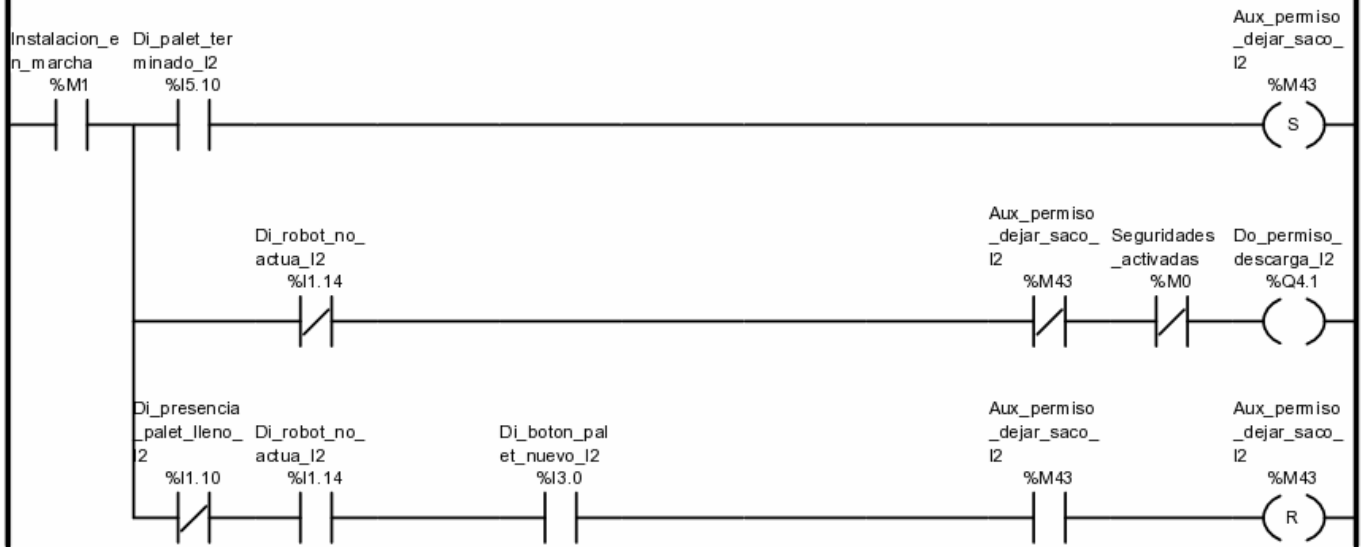
Fecha: 28/8/09

11.2.5 LD_Mast_Permisos.



(*permiso para descargar en la mesa de paletizado 2 *)

TOP+2

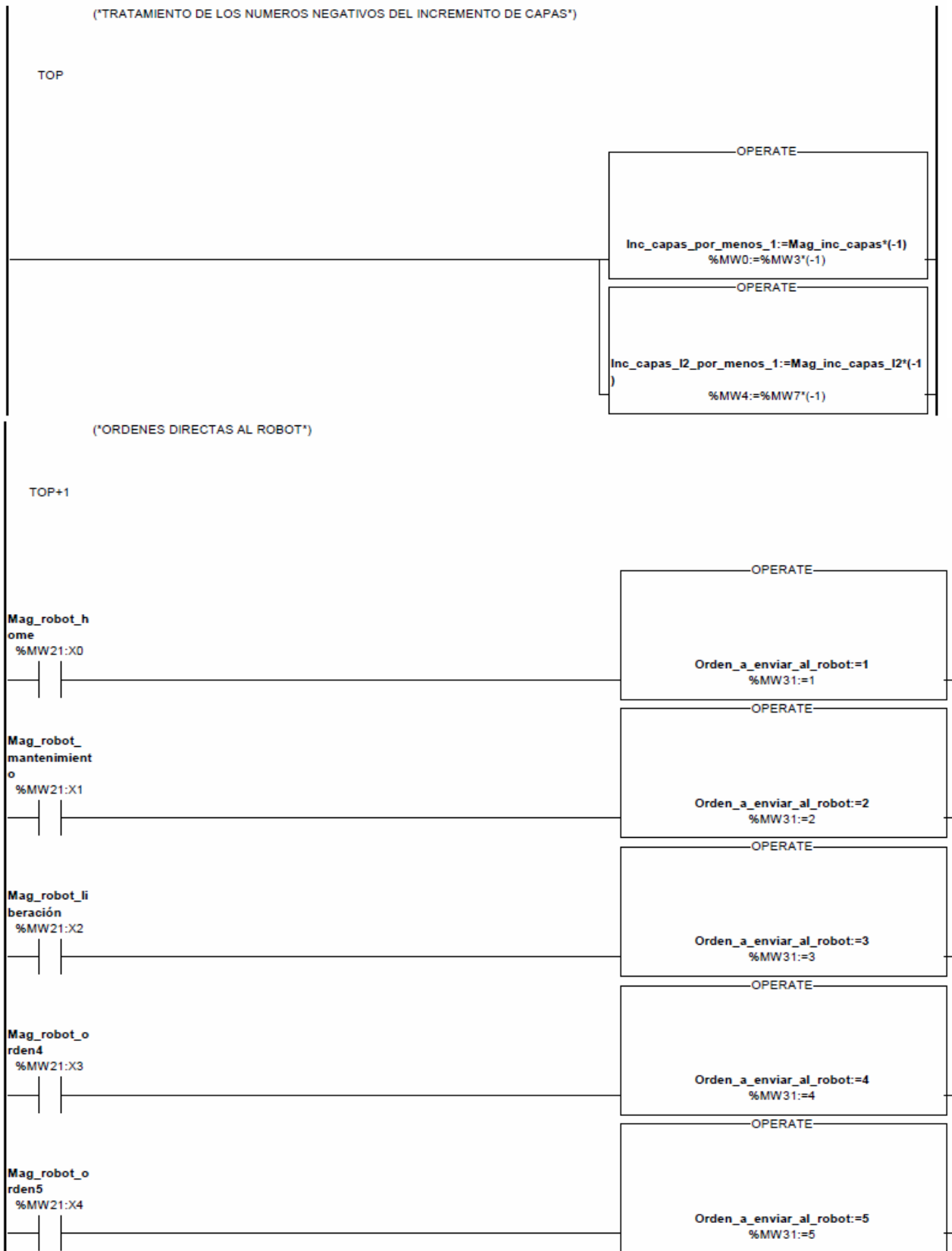


Carpeta: 2

MAST - Permisos

Fecha: 28/8/09

11.2.6 LD_Mast_Com_Robot.



Mag_robot_o
rden6
%MW21:X5

Mag_robot_o
rden7
%MW21:X6

OPERATE

Orden_a_enviar_al_robot:=6
%MW31:=6

OPERATE

Orden_a_enviar_al_robot:=7
%MW31:=7

(*ORDENES DIRECTAS AL ROBOT*)

TOP+2

Mag_robot_o
rden8
%MW21:X7

Mag_robot_o
rden9
%MW21:X8

Mag_robot_o
rden10
%MW21:X9

Mag_robot_o
rden11
%MW21:X10

Mag_robot_o
rden12
%MW21:X11

OPERATE

Orden_a_enviar_al_robot:=8
%MW31:=8

OPERATE

Orden_a_enviar_al_robot:=9
%MW31:=9

OPERATE

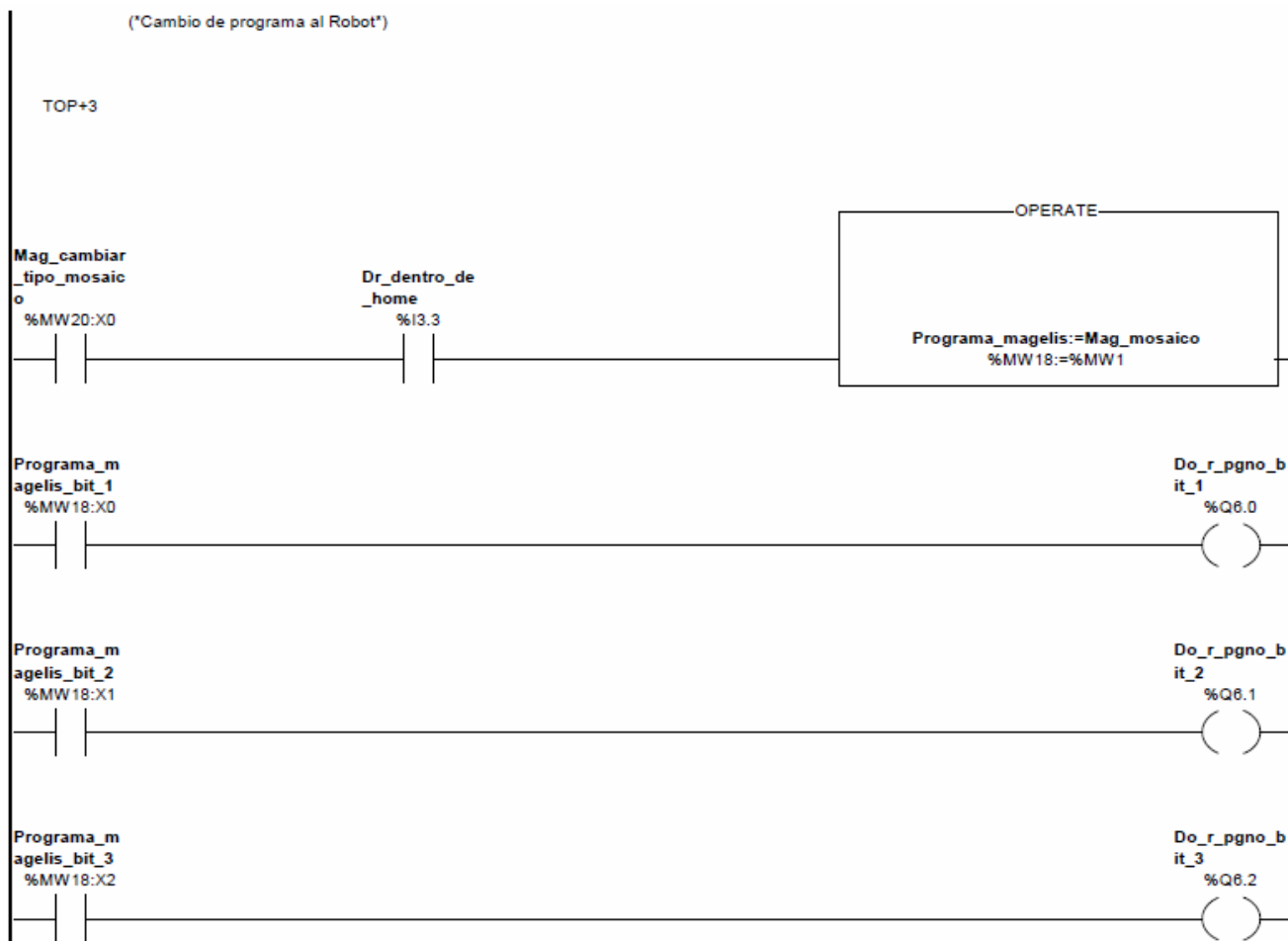
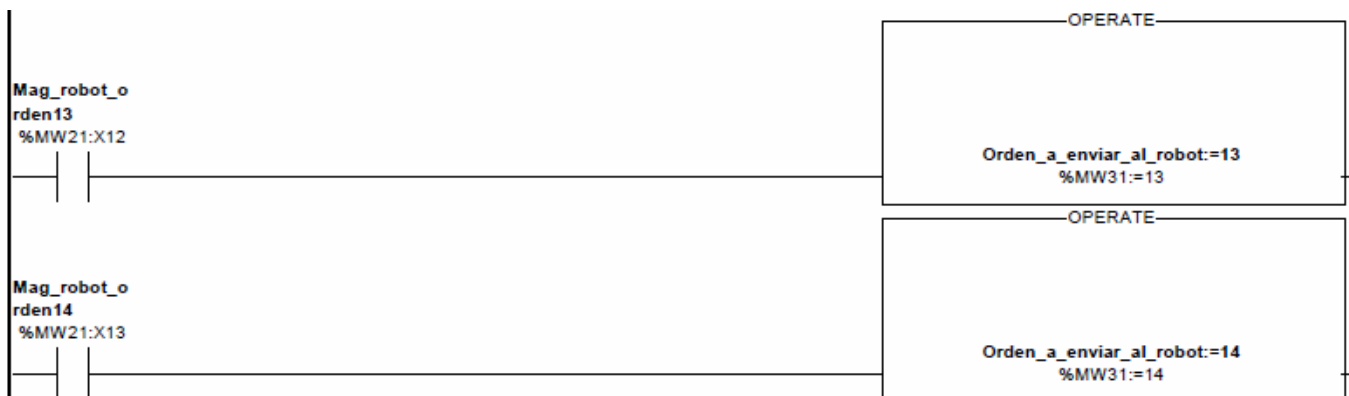
Orden_a_enviar_al_robot:=10
%MW31:=10

OPERATE

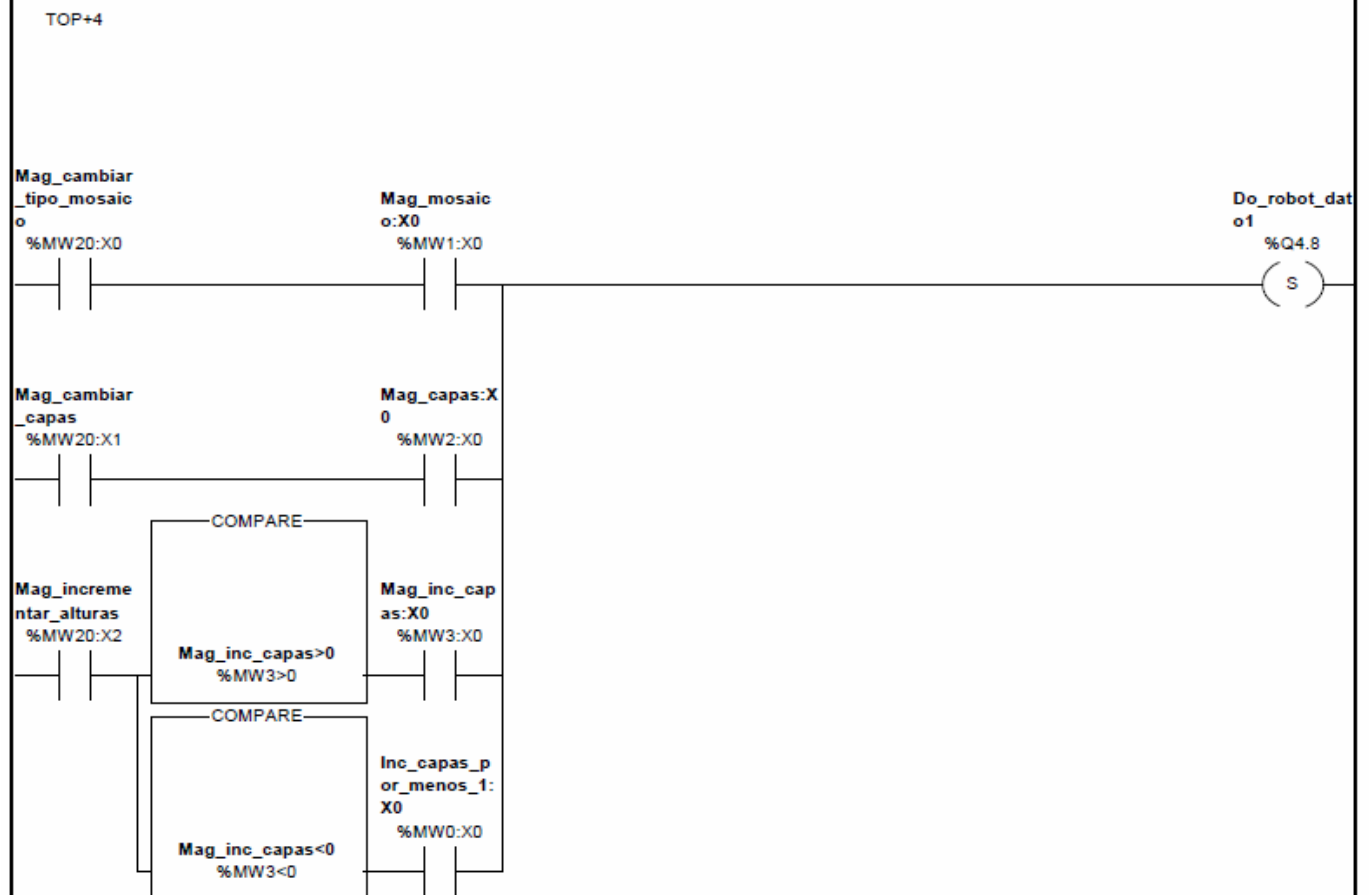
Orden_a_enviar_al_robot:=11
%MW31:=11

OPERATE

Orden_a_enviar_al_robot:=12
%MW31:=12

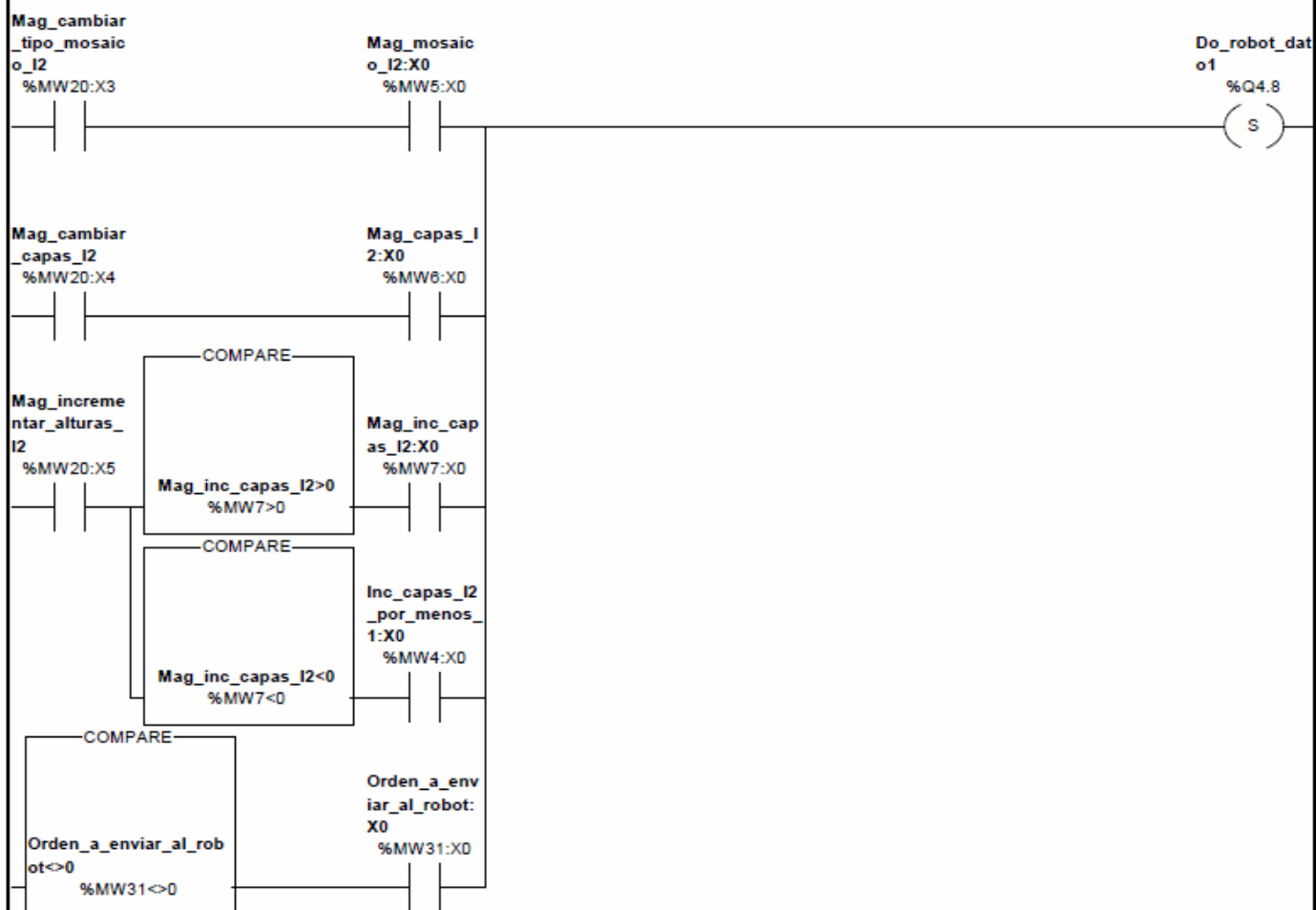


(*SALIDAS DE DATO 1 DE COMUNICACION CON EL ROBOT*)



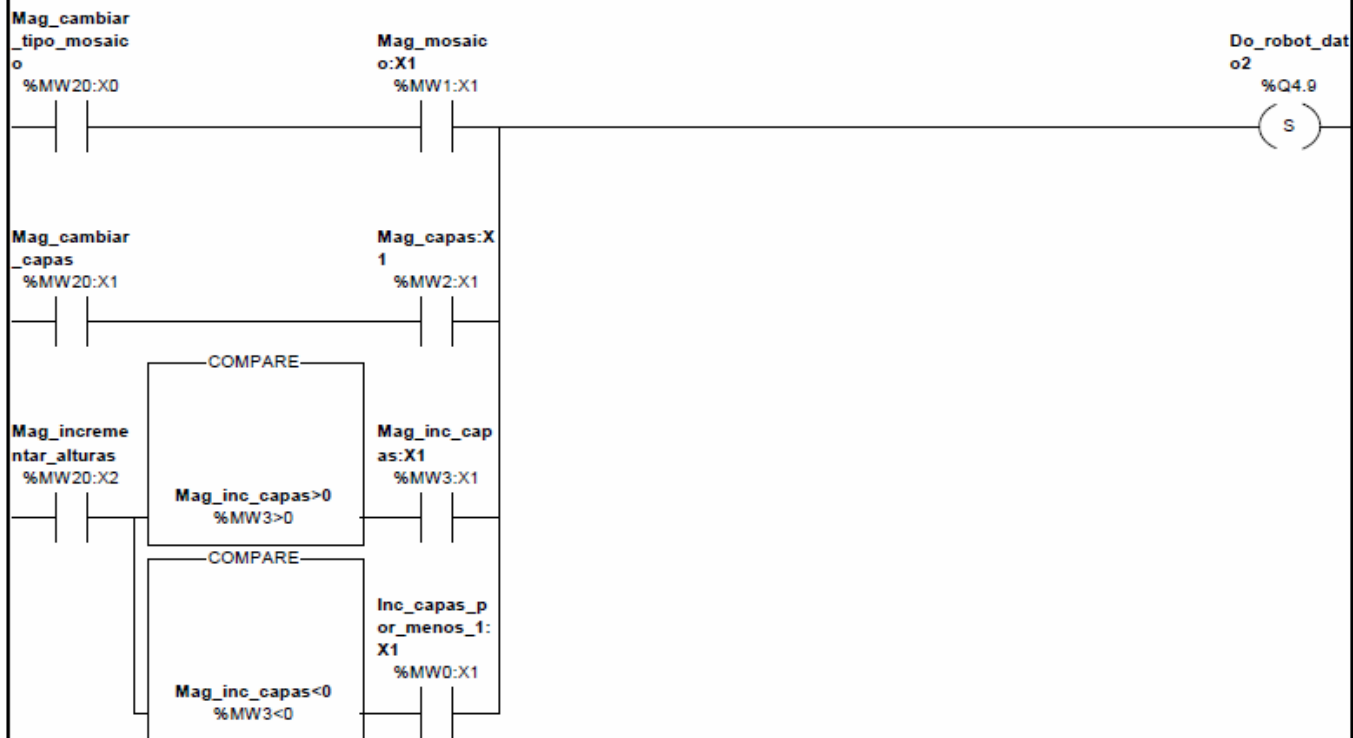
("SALIDAS DE DATO 1 DE COMUNICACION CON EL ROBOT")

TOP+5

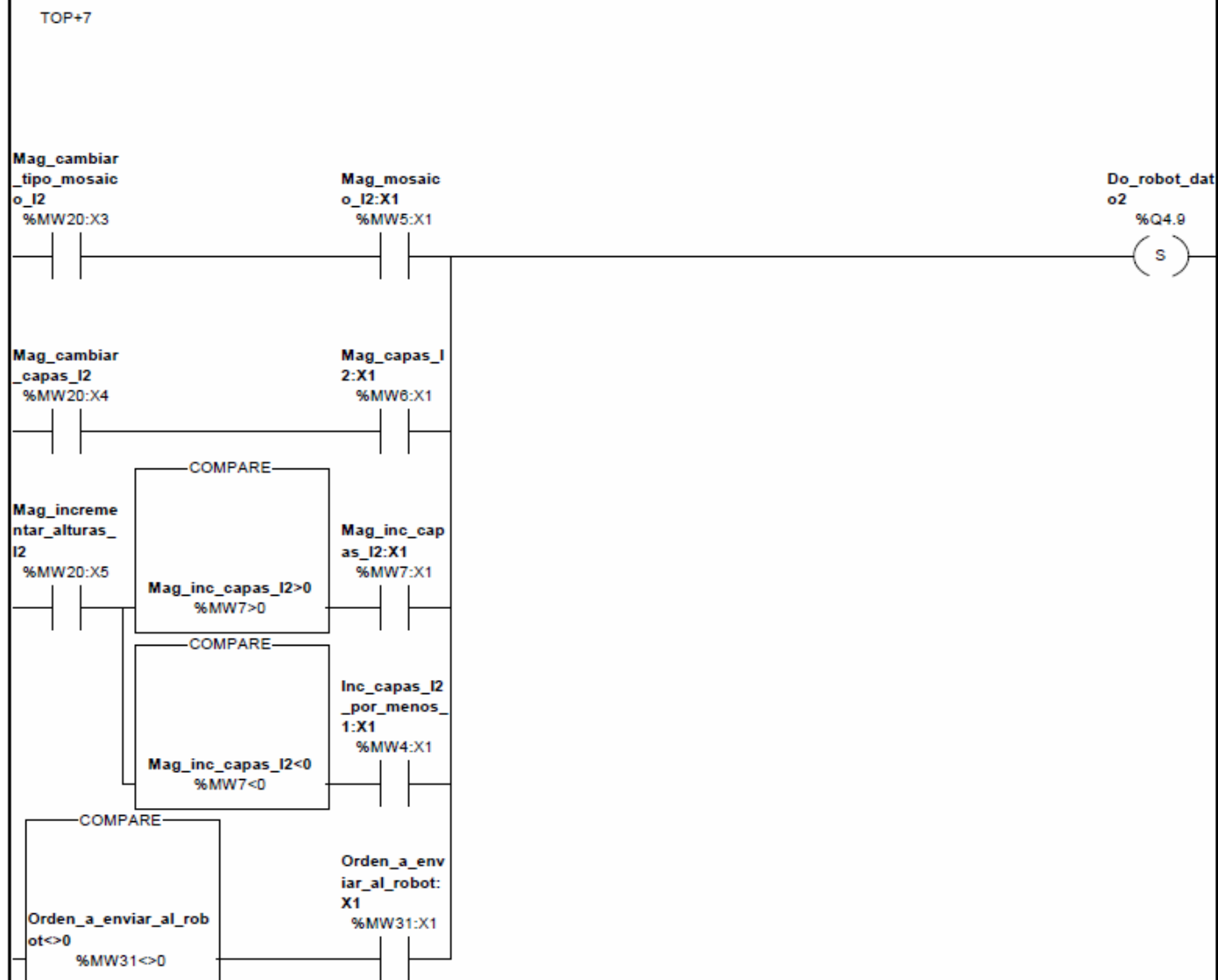


(*SALIDAS DE DATO 2 DE COMUNICACION CON EL ROBOT*)

TOP+8

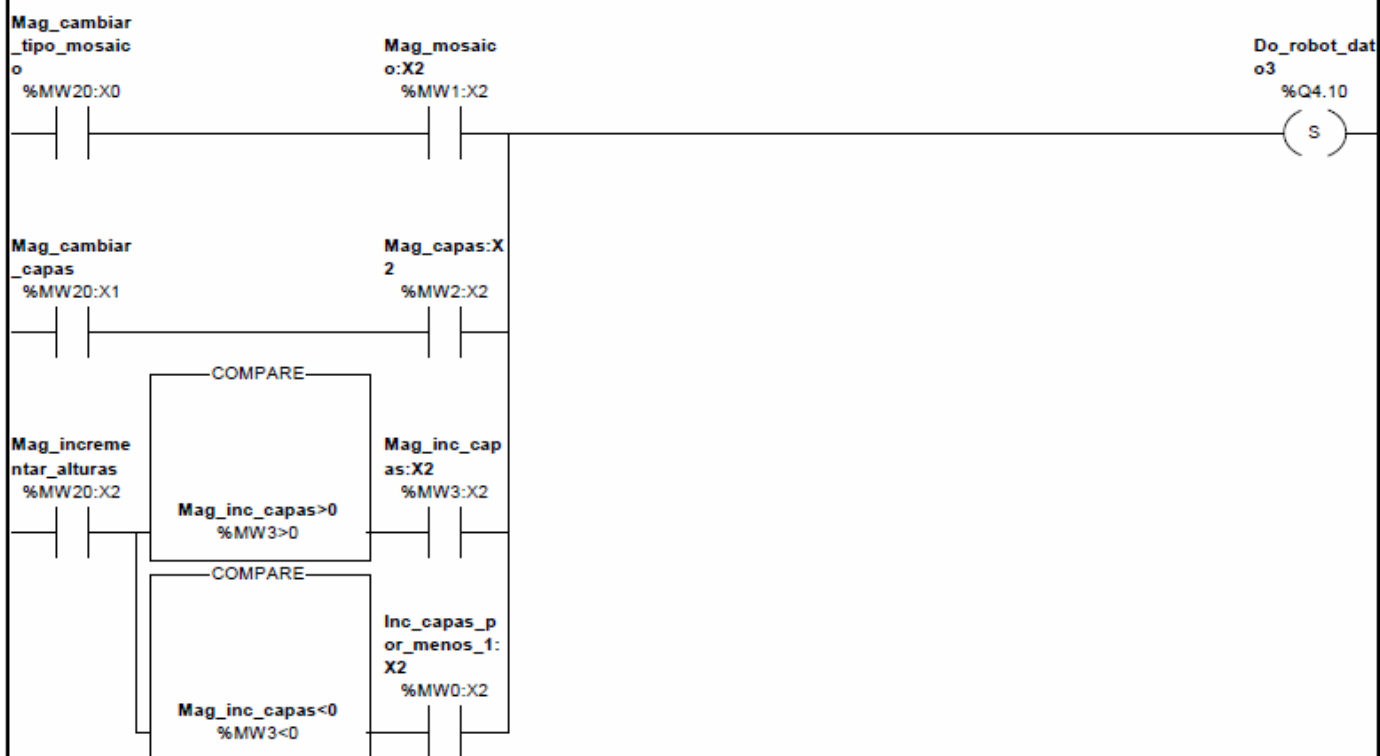


(*SALIDAS DE DATO 2 DE COMUNICACION CON EL ROBOT*)



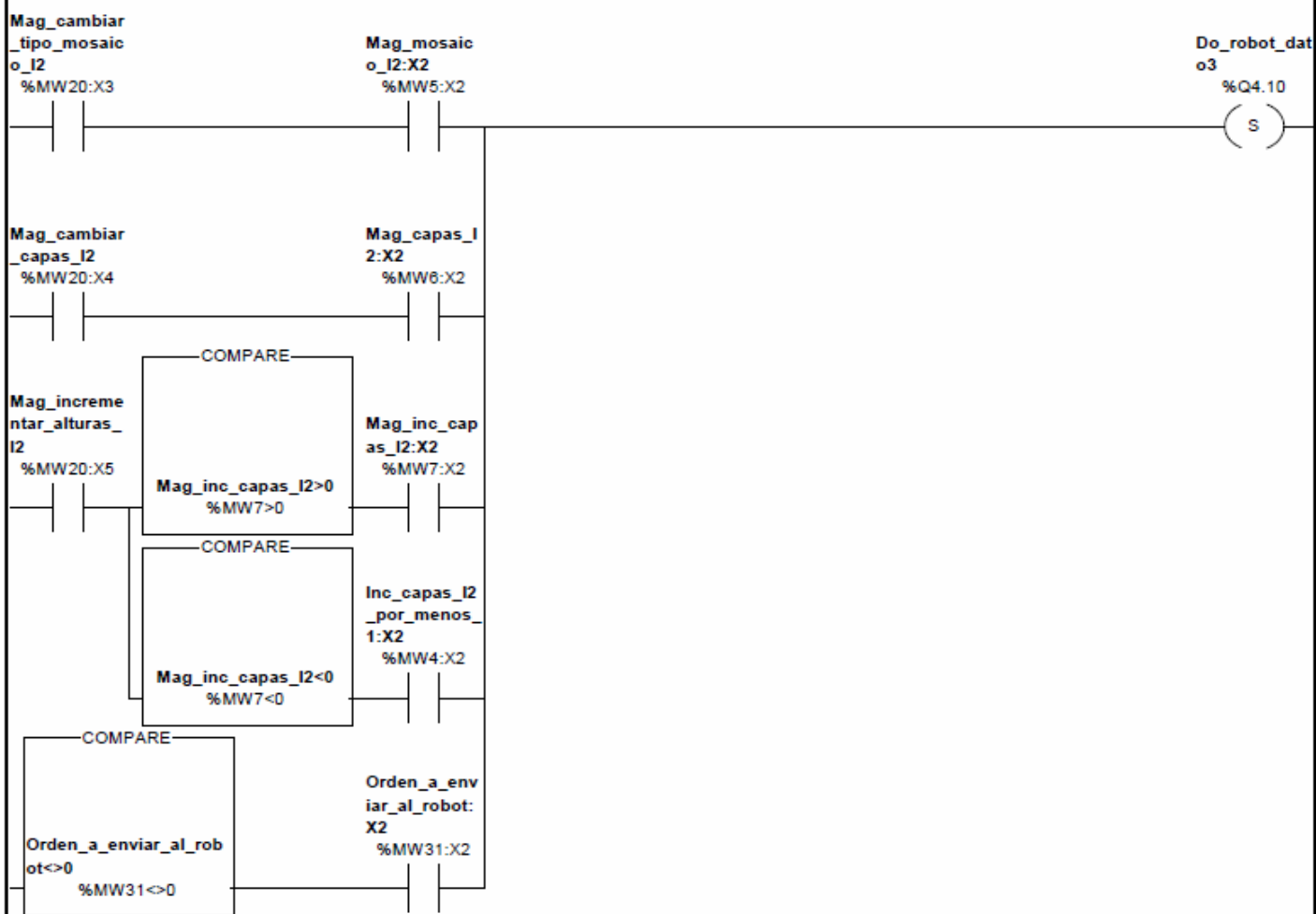
(*SALIDAS DE DATO 3 DE COMUNICACION CON EL ROBOT*)

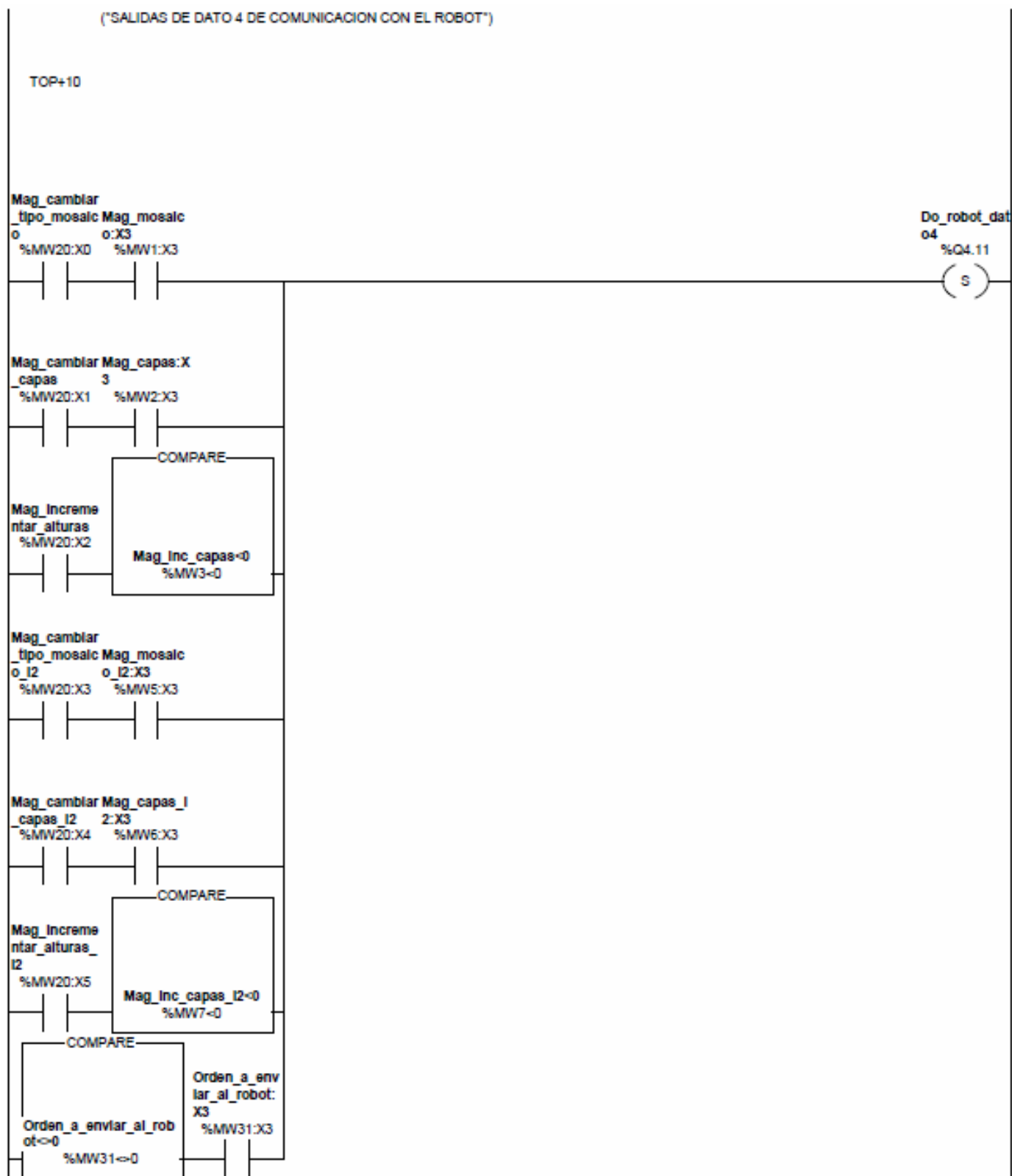
TOP+8



(*SALIDAS DE DATO 3 DE COMUNICACION CON EL ROBOT*)

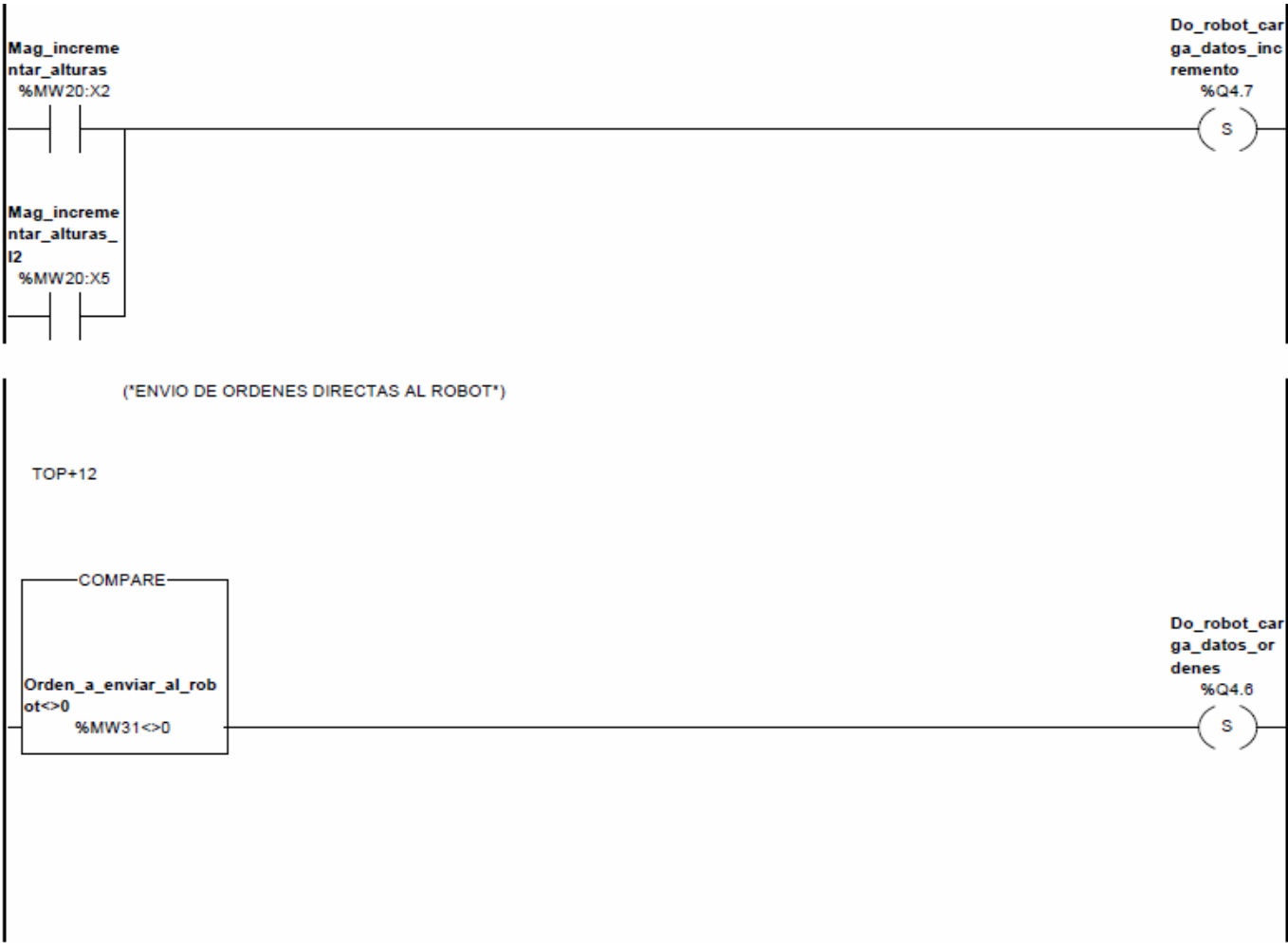
TOP+9





(*CLASIFICACION DE LOS DATOS QUE SE VAN A ENVIAR*)





(*EXISTE ALGUN DATO QUE SE QUIERE ENVIAR AL ROBOT*)

TOP+13

Mag_cambiar
tipo_mosaico
%MW20:X0

Se_va_a_enviar_dato_I1_a_robot
%M11

(S)

Mag_cambiar
capas
%MW20:X1

Mag_incrementar_alturas
%MW20:X2

COMPARE

Orden_a_enviar_al_robot
ot<>0
%MW31<>0

Mag_cambiar
tipo_mosaico_I2
%MW20:X3

Se_va_a_enviar_dato_I2_a_robot
%M12

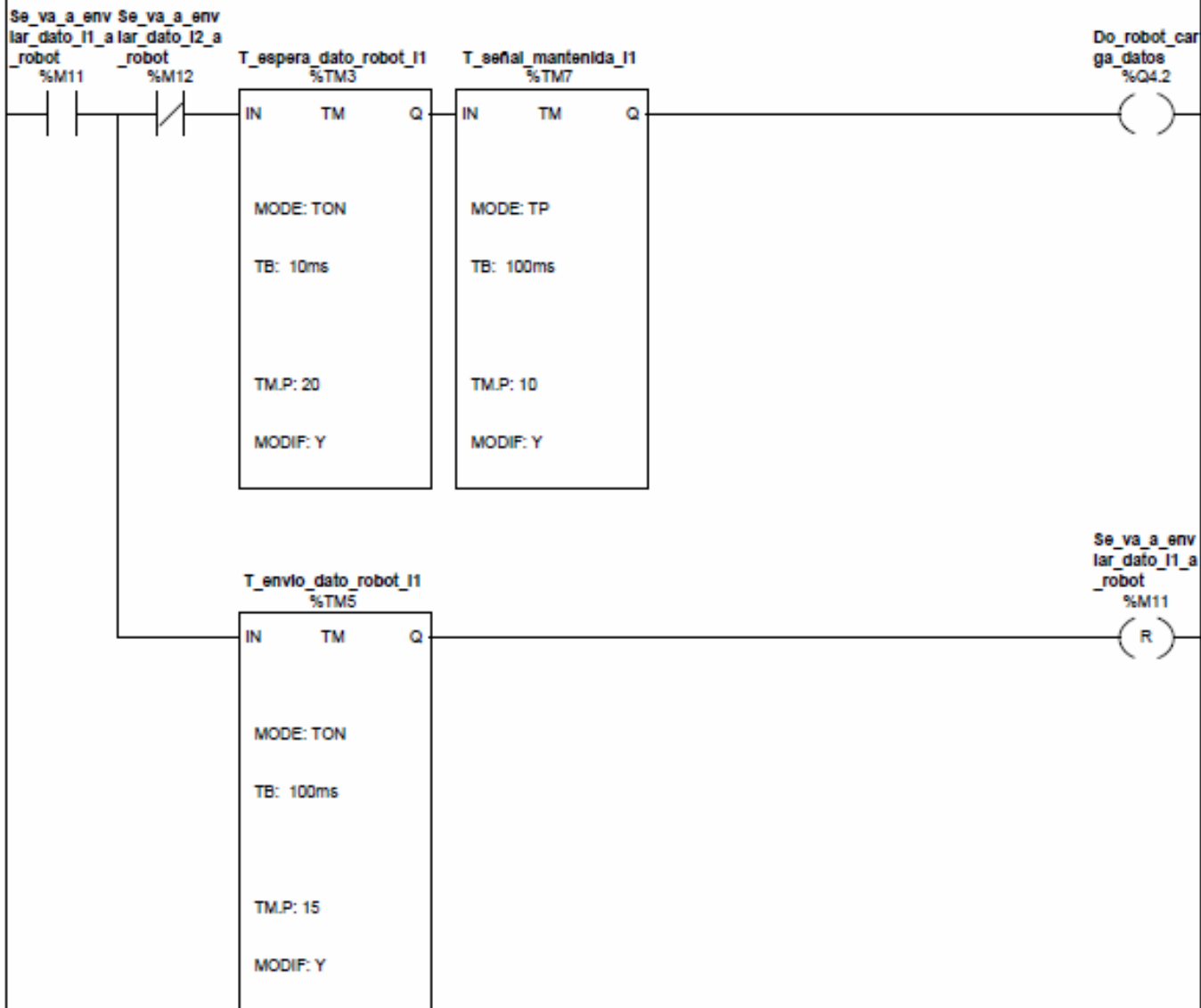
(S)

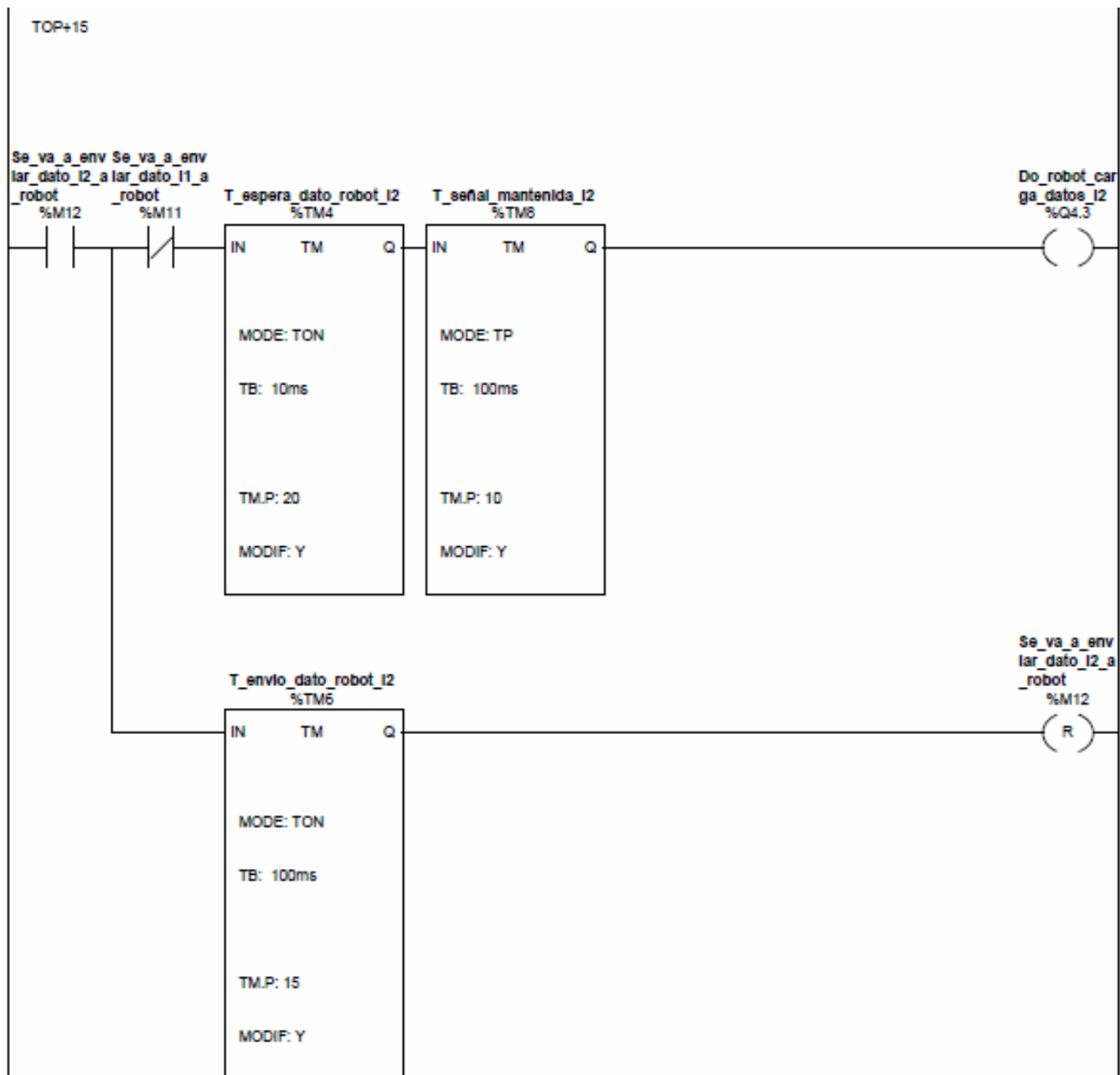
Mag_cambiar
capas_I2
%MW20:X4

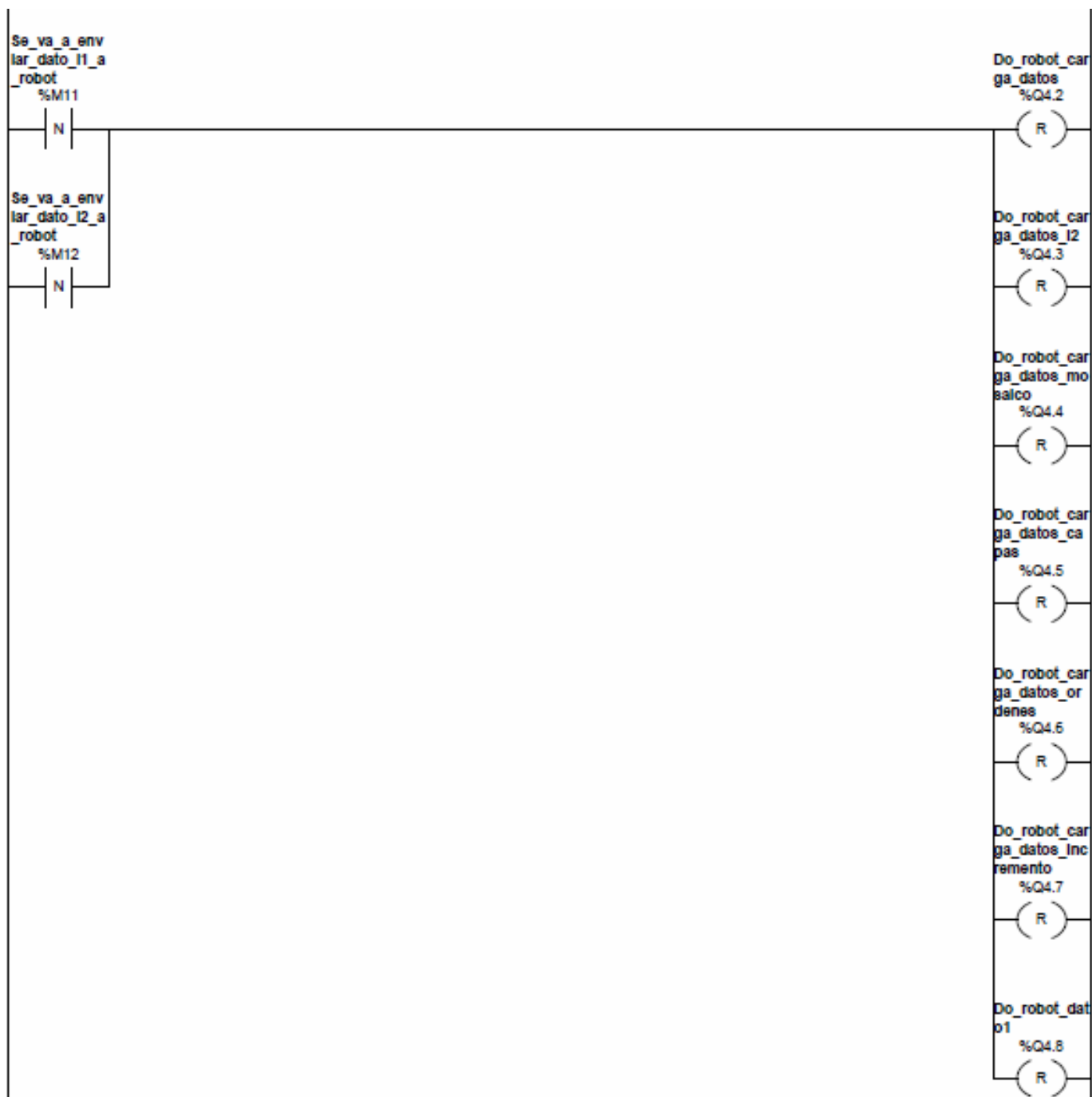
Mag_incrementar_alturas_I2
%MW20:X5

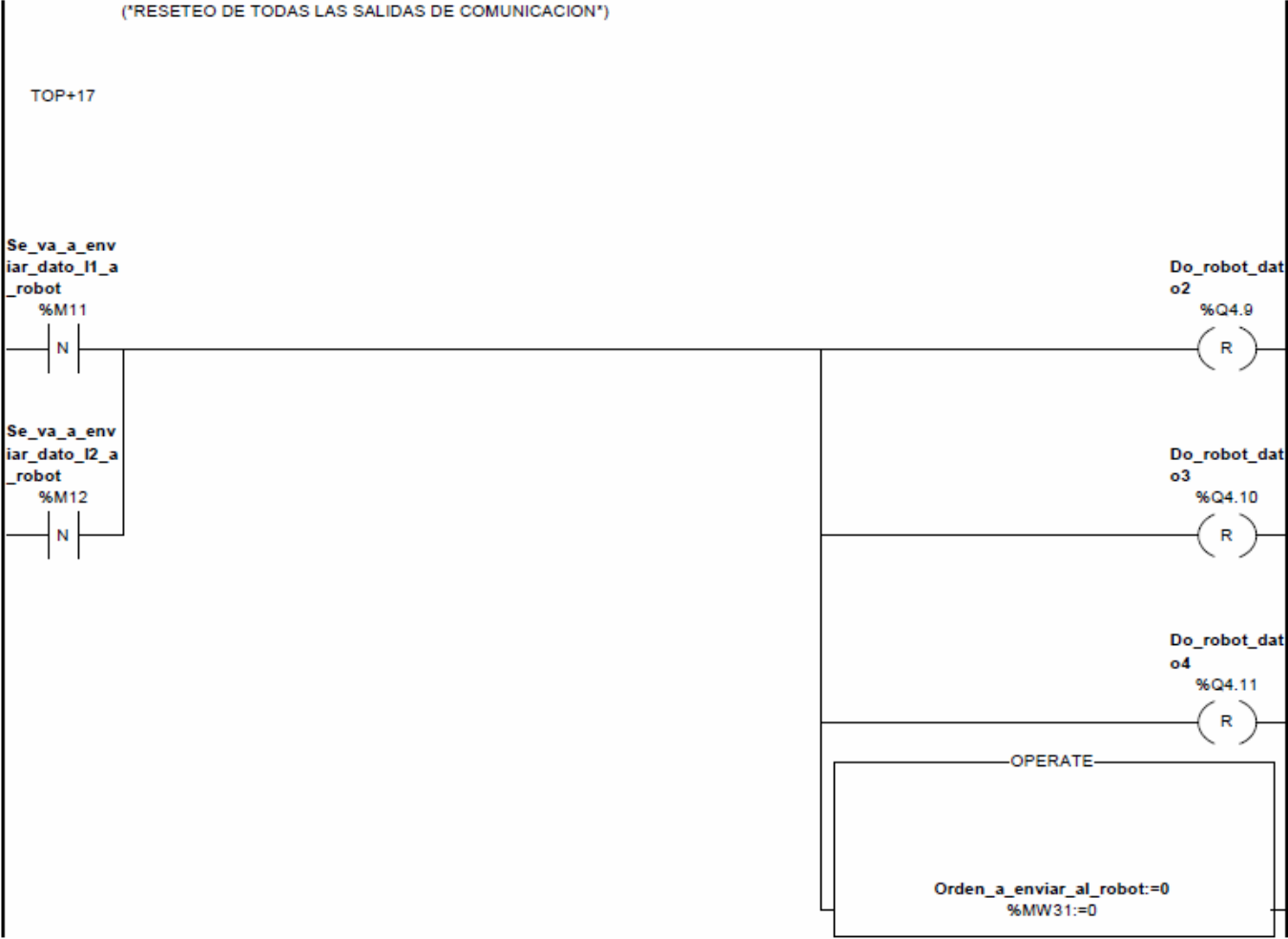
("ESPERA A QUE EL ROBOT PROCESA LAS SALIDAS QUE SE ENVIAN")



TOP+14



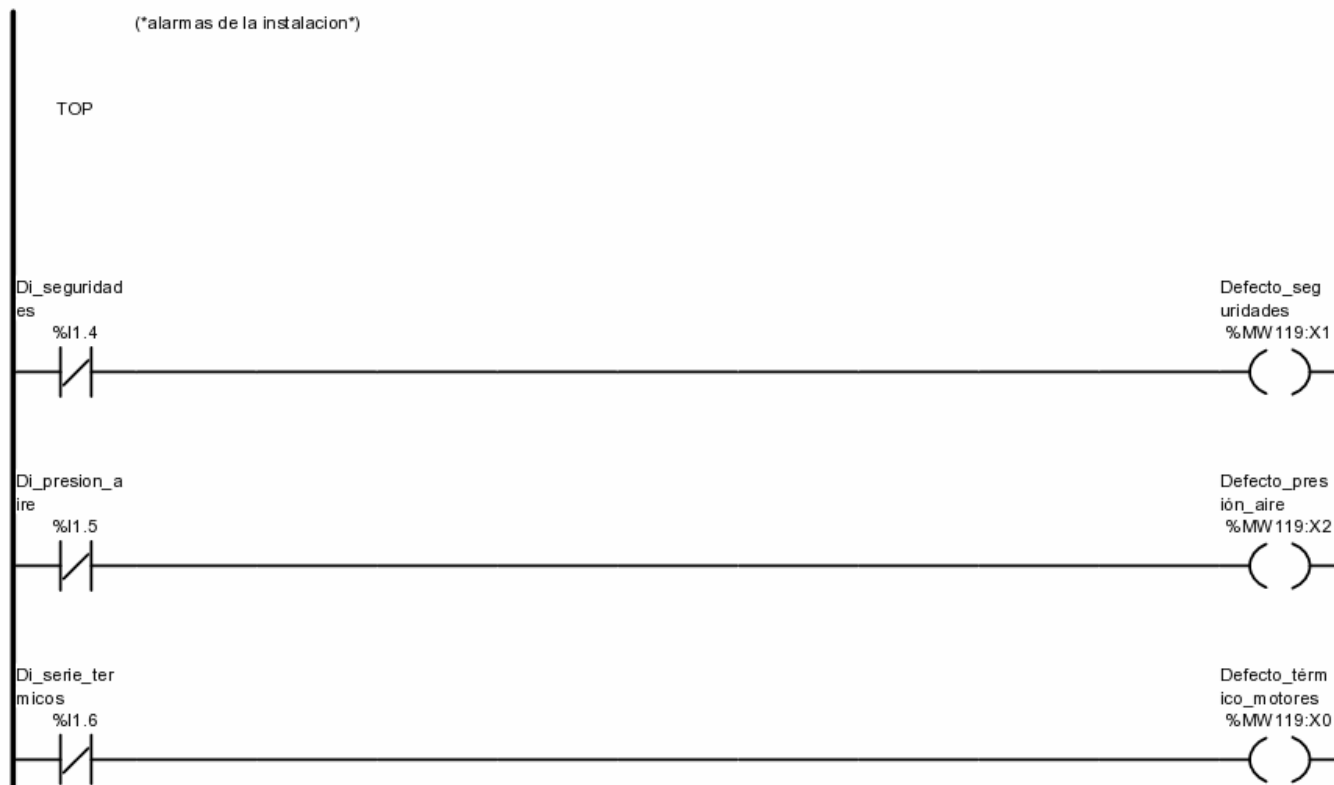






	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

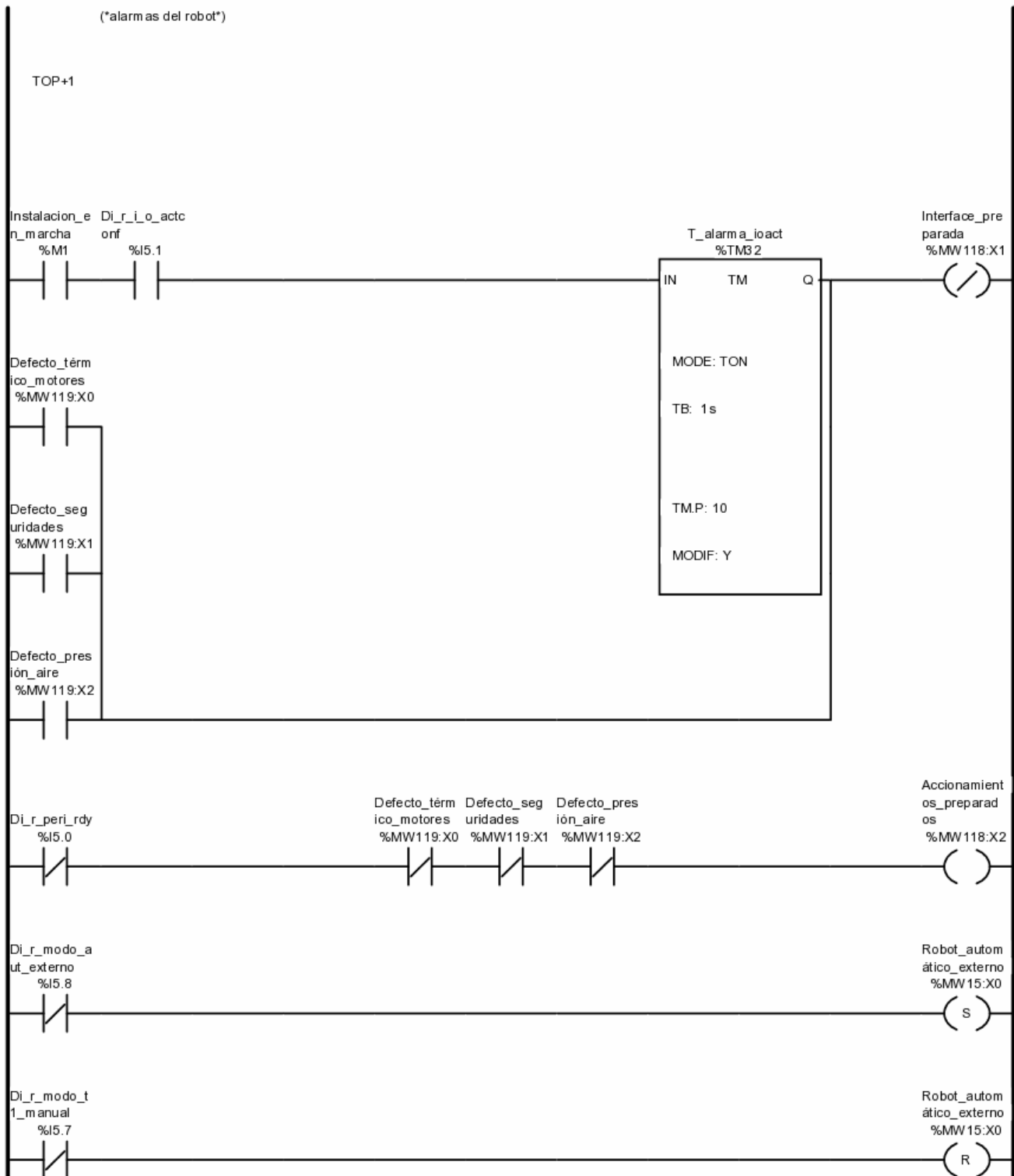
11.2.7 LD_Mast_Alarmas.

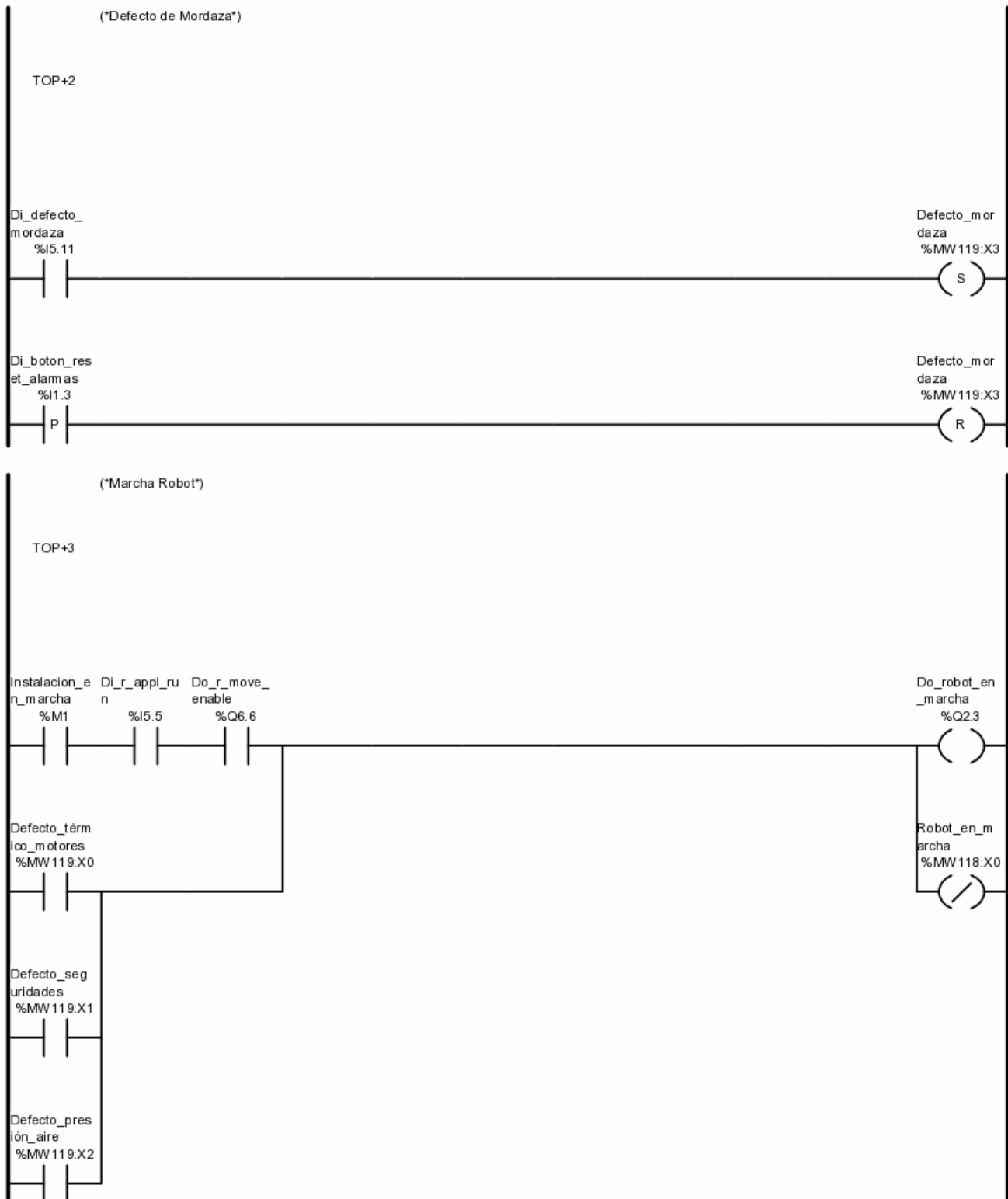


Carpeta: 1

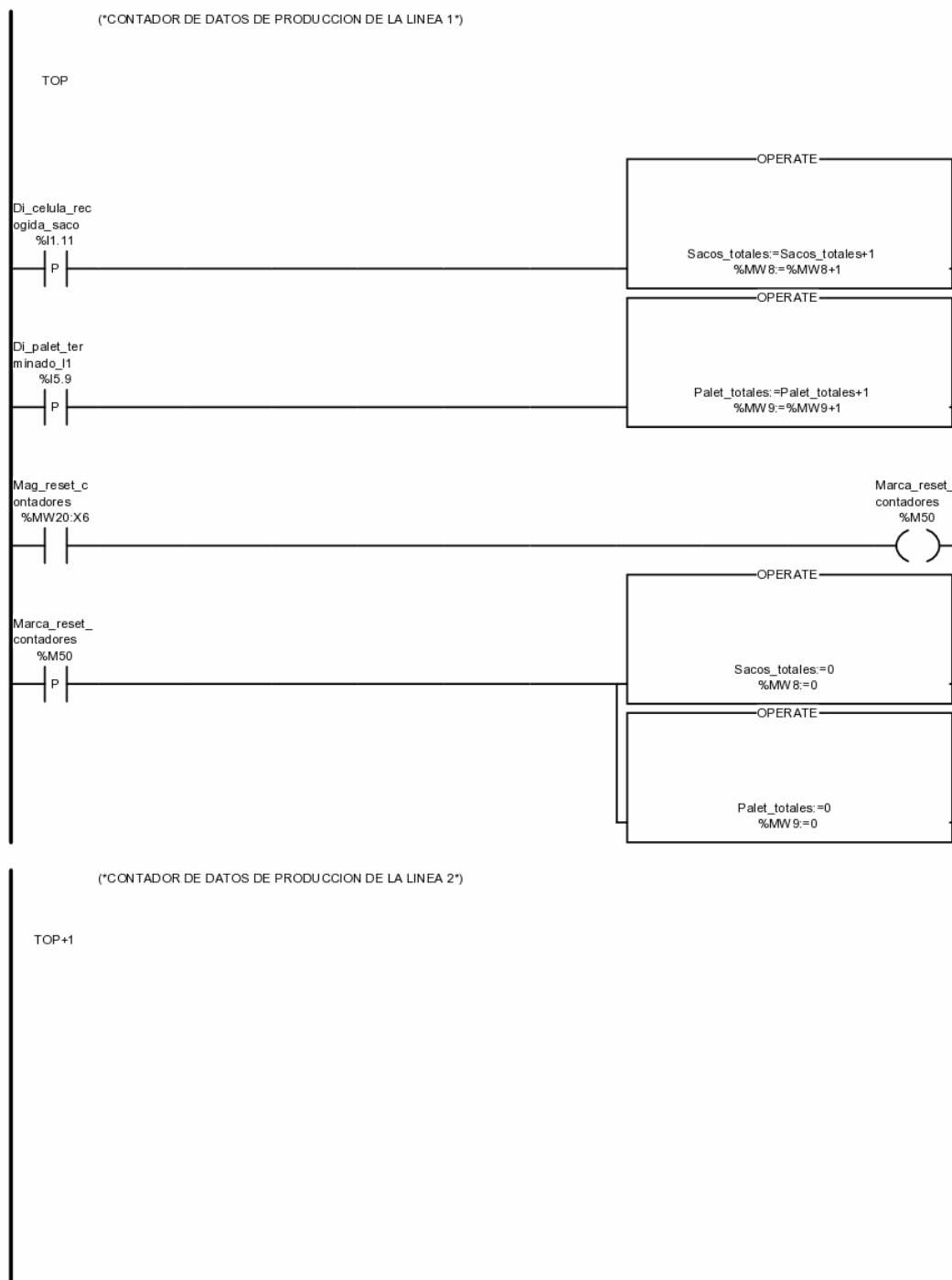
MAST - Alarmas

Fecha: 28/8/09





11.2.8 LD_Mast_Producción.



12 ANEXO.E: (Programa de Robot Paletizado).

12.1 MOSAICO_3.src / MOSAICO_4.src. / MOSAICO_5.src.

```

DEF mosaico_3( )
ext zona_222 ( )
ext zona_111 ( )
ext coi_sgrd ( )
ext mantenimiento ( )
ext home ( )

INI
;Make your modifications here

;=====PUNTOS DE SEGURIDAD=====
coi_sgrd ( )
$out[21]=false

;=====Reset de mordaza=====
pulse ($out[17],true,2)
pulse ($out[19],true,2)
$timer[1]=0
saco_1=1
saco_2=1
base_data[3]=base_data[1]
base_data[4]=base_data[2]
$out[30]=false

;=====PROGRAMA CICLICO DEPENDIENDO DE SELECCION=====
loop

continue
if (($out[6]==false) and ($in[7]==true) and ($out[9]==true)) then
$out[6]=true
endif

;=====RESTOS=====

continue
if $out[25]==true then
wait sec 1
pulse ($out[10],true,2)
pulse ($out[11],true,2)
saco_1=1
saco_2=1
base_data[3]=base_data[1]
base_data[4]=base_data[2]
wait sec 3
$out[25]=false
endif

;=====MANTENIMIENTO=====
continue
if (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false) and
(registro_1==2)) then
mantenimiento()
endif

;=====ZONA PALETIZADO 1=====
continue
if $out[25]==false then
continue
if (($in[11]==true) and ($in[13]==true) and ($out[30]==false)) then
zona_111 ( )

```

```

DEF mosaico_3() = DEF mosaico_4 ( ) = DEF mosaico_5 ( )
zona_222() = zona_22 ( ) = zona_2 ( )
zona_111() = zona_11 ( ) = zona_1 ( )

```

Mosaico_4() y
Mosaico_5() sería
igual pero cambiando
las variables.

```

endif
endif

;=====ZONA PALETIZADO 2=====
continue
if $out[25]==false then
continue
if (($in[11]==true) and ($in[14]==true) and ($out[30]==true)) then
zona_222 ()
endif
endif

;=====HOME=====
=
continue
if (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false) and
(registro_1==1) and (saco_1<2) and (saco_2<2)) then
home ()
endif

;=====SALIDA CONDICIONAL=====
continue
if (((program==1) or (program==2)) and ($in[11]==false) and (saco_1<2) and (saco_2<2))
then
wait sec 1
exit
endif

endloop
wait sec 0
coi_sgrd ()

END

```

MOSAICO_5 () -> if (((program==2) or (program==3))
MOSAICO_4 () -> if (((program==1) or (program==3))

12.2 MANTENIMIENTO.src

```

DEF mantenimiento( )
INI
;Make your modifications here

coi_sgrd ()

PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P2 CONT Vel= 30 % PDAT2 Tool[1] Base[0]
PTP P3 CONT Vel= 30 % PDAT3 Tool[1] Base[0]
;punto de mantenimiento
PTP P4 CONT Vel= 30 % PDAT4 Tool[1] Base[0]

;=====salida de mantenimiento=====



wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false)
and (registro_1==3))

PTP P5 CONT Vel= 30 % PDAT5 Tool[1] Base[0]

coi_sgrd ()

END

```

	Proyecto PROYECTO FIN DE CARRERA	
	Documento Instalación de Robot Paletizado.	

12.3 HOME.src

```

DEF home( )

coi_sgrd ( )

;marca para saber que estamos dentro del Home y cambiar de programa
$out[21]=true

wait for (($in[17]==false) and ($in[18]==false) and ($in[19]==true) and ($in[20]==false)
and (registro_1==3))

;=====CAMBIO DE MOSAICOS=====

;Mosaico seleccionado
$out[21]=false

;=====CAMBIO DE CAPAS=====

capas_1=reg_capas_1
if program==1 then
sacos_1=capas_1*5
sacos_2=capas_1*5
endif
if program==2 then
sacos_1=capas_1*4
sacos_2=capas_1*4
endif
if program==3 then
sacos_1=capas_1*3
sacos_2=capas_1*3
endif

END

```

12.4 COI_SGRD.src

```

DEF coi_sgrd( )
INI
;Make your modifications here

;=====ENCIMA MESA DE PALETIZADO LINEA 2=====
if ($axis_act.a1>60) then
PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[0]
PTP P3 CONT Vel= 100 % PDAT3 Tool[1] Base[0]
endif
;=====TRANSITO PALET_2 ZONA RECOGIDA=====
if (($axis_act.a1<=60) and ($axis_act.a1>=-35)) then
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[0]
endif
;=====TRANSITO LINEA_1 y ZONA RECOGIDA=====
if (($axis_act.a1<=-50) and ($axis_act.a1>=-80)) then
PTP P7 CONT Vel= 100 % PDAT7 Tool[1] Base[0]
endif
;=====ENCIMA MESA PALETIZADO LINEA 1 =====
if ($axis_act.a1<=-80) then
PTP P8 CONT Vel= 100 % PDAT8 Tool[1] Base[0]
endif
;=====ENCIMA MESA RECOGIDA=====
if (($axis_act.a1>=-50) and ($axis_act.a1<=-35)) then
PTP HOME CONT Vel= 100 % PDAT17

```


endif

PTP HOME CONT Vel= 100 % PDAT19

;punto por encima de Home

PTP P4 CONT Vel= 100 % PDAT4 Tool[1] Base[0]

;punto por debajo de Home

PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]

PTP HOME CONT Vel= 100 % PDAT18

END

12.5 ZONA_1.src / ZONA_2.src

DEF zona_1()

alt_incr_1=190

\$out[30]=true

coger_1()

;;;saco 1 capa 1;;;;;

if (saco_1==1) or (saco_1==11) or (saco_1==21) or (saco_1==31) or (saco_1==41) or (saco_1==51) or (saco_1==61) or (saco_1==71) then

PTP P1 CONT Vel= 100 % PDAT3 Tool[1] Base[3]

ptp_rel {z -140} c_ptp

pulse (\$out[19],true,2)

wait sec 0.3

endif

;;;;;;;;saco 1 capa 2;;;;;;;;;

if (saco_1==6) or (saco_1==16) or (saco_1==26) or (saco_1==36) or (saco_1==46) or (saco_1==56) or (saco_1==66) then

PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[3]

ptp_rel {z -140} c_ptp

pulse (\$out[19],true,2)

wait sec 0.3

endif

;;;;;;;;saco 2 capa 1;;;;;;;;;

if (saco_1==2) or (saco_1==12) or (saco_1==22) or (saco_1==32) or (saco_1==42) or (saco_1==52) or (saco_1==62) or (saco_1==72) then

PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[3]

ptp_rel {z -140} c_ptp

pulse (\$out[19],true,2)

wait sec 0.3

endif

;;;;;;;;saco 2 capa 2;;;;;;;;;

if (saco_1==7) or (saco_1==17) or (saco_1==27) or (saco_1==37) or (saco_1==47) or (saco_1==57) or (saco_1==67) then

PTP P7 CONT Vel= 100 % PDAT7 Tool[1] Base[3]

ptp_rel {z -140} c_ptp

pulse (\$out[19],true,2)

wait sec 0.3

endif

;;;;;;;;saco 3 capa 1;;;;;;;;;

Zona_1.src = Zona_2.src
DEF zona_1() = DEF zona_2 ().
Coger_1 () = Coger_2 ().
Saco_1 = saco_2.
Base_data[3] = base_data[1].
Base_data[4] = base_data[2].
Sacos_1 = Sacos_2.
pulse_out[10] = pulse_out[11].
PTP->Tool[1] Base[3] = PTP->Tol[1] Base[4]

Zona_2.src sería igual que Zona_1.src pero cambiando las variables.



```
if (saco_1==3) or (saco_1==13) or (saco_1==23) or (saco_1==33) or (saco_1==43) or
(saco_1==53) or (saco_1==63) or (saco_1==73) then
PTP P3 CONT Vel= 100 % PDAT4 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 3 capa2;;;;;;;;;;;;

if (saco_1==8) or (saco_1==18) or (saco_1==28) or (saco_1==38) or (saco_1==48) or
(saco_1==58) or (saco_1==68) then
PTP P8 CONT Vel= 100 % PDAT8 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 4 capa 1;;;;;;;;;;;;

if (saco_1==4) or (saco_1==14) or (saco_1==24) or (saco_1==34) or (saco_1==44) or
(saco_1==54) or (saco_1==64) or (saco_1==74) then
PTP P4 CONT Vel= 100 % PDAT5 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 4 capa 2;;;;;;;;;;;;

if (saco_1==9) or (saco_1==19) or (saco_1==29) or (saco_1==39) or (saco_1==49) or
(saco_1==59) or (saco_1==69) then
PTP P9 CONT Vel= 100 % PDAT9 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 5 capa 1;;;;;;;;;;;;

if (saco_1==5) or (saco_1==15) or (saco_1==25) or (saco_1==35) or (saco_1==45) or
(saco_1==55) or (saco_1==65) or (saco_1==75) then
PTP P5 CONT Vel= 100 % PDAT10 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 5 capa 2;;;;;;;;;;;;

if (saco_1==10) or (saco_1==20) or (saco_1==30) or (saco_1==40) or (saco_1==50) or
(saco_1==60) or (saco_1==70) then
PTP P10 CONT Vel= 100 % PDAT11 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;permiso para sacar palet;;;;;;;;;;;;
wait sec 0
if (saco_1==sacos_1) then
pulse ($out[10],true,2)
saco_1=0
base_data[3]=base_data[1]
wait sec 0.5
endif
```

;;;;;;;;;incremento saco;;;;;;;;;

saco_1=saco_1+1

ptp_rel{z 110}c_ptp

;;;;;;;;;punto elevado encima de zona 2

SYN PULSE 17 'Abrir_Manos' State= TRUE Time= 0.5 sec at START Delay= 0 ms

PTP P16 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

PTP P11 CONT Vel= 100 % PDAT1 Tool[1] Base[0]

PTP P24 CONT Vel= 100 % PDAT24 Tool[1] Base[0]

PTP P25 CONT Vel= 100 % PDAT25 Tool[1] Base[0]

;verificacion manos y dedos abiertos

continue

loop

continue

if ((\$in[27]==false) or (\$in[28]==false)) then

\$timer_stop[1]=false

endif

continue

if \$timer[1]>=4000 then

wait sec 0

\$timer_stop[1]=true

\$timer[1]=0

pulse (\$out[12],true,2)

endif

continue

if ((\$in[27]==true) and (\$in[28]==true)) then

\$timer_stop[1]=true

\$timer[1]=0

exit

endif

endloop

END

12.6 ZONA_11.src / ZONA_22-src.

DEF zona_11()

alt_incr_1=200

\$out[30]=true

coger_1()

;;;;;;;;saco 1 capa 1;;;;;;;;;

if (saco_1==1) or (saco_1==9) or (saco_1==17) or (saco_1==25) or (saco_1==33) or (saco_1==41) or (saco_1==49) or (saco_1==57) then

PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[3]

ptp_rel {z -140} c_ptp

pulse (\$out[19],true,2)

wait sec 0.3

endif

;;;;;;;;saco 1 capa 2;;;;;;;;;

if (saco_1==5) or (saco_1==13) or (saco_1==21) or (saco_1==29) or (saco_1==37) or (saco_1==45) or (saco_1==53) then

PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[3]

ptp_rel {z -120} c_ptp

Zona_11.src = Zona_22.src
DEF zona_11() = DEF zona_22 ().
Coger_1 () = Coger_2 ().
Saco_1 = saco_2.
Base_data[3] = base_data[1].
Base_data[4] = base_data[2].
Sacos_1 = Sacos_2.
pulse_out[10] = pulse_out[11].

Zona_22.src
sería igual que
Zona_11.src
pero cambiando
las variables.

```

pulse ($out[19],true,2)
wait sec 0.3

endif

;;;;;;;;;;;;saco 2 capa 1;;;;;;;;;;;;

if (saco_1==3) or (saco_1==11) or (saco_1==19) or (saco_1==27) or (saco_1==35) or
(saco_1==43) or (saco_1==51) or (saco_1==59) then
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 2 capa 2;;;;;;;;;;;;

if (saco_1==7) or (saco_1==15) or (saco_1==23) or (saco_1==31) or (saco_1==39) or
(saco_1==47) or (saco_1==55) then
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 3 capa 1;;;;;;;;;;;;

if (saco_1==2) or (saco_1==10) or (saco_1==18) or (saco_1==26) or (saco_1==34) or
(saco_1==42) or (saco_1==50) or (saco_1==58) then
PTP P3 CONT Vel= 100 % PDAT3 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 3 capa2;;;;;;;;;;;;

if (saco_1==6) or (saco_1==14) or (saco_1==22) or (saco_1==30) or (saco_1==38) or
(saco_1==46) or (saco_1==54) then
PTP P7 CONT Vel= 100 % PDAT7 Tool[1] Base[3]
ptp_rel {z -120} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif

;;;;;;;;;;;;saco 4 capa 1;;;;;;;;;;;;

if (saco_1==4) or (saco_1==12) or (saco_1==20) or (saco_1==28) or (saco_1==36) or
(saco_1==44) or (saco_1==52) or (saco_1==60) then
PTP P4 CONT Vel= 100 % PDAT4 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 4 capa 2;;;;;;;;;;;;

if (saco_1==8) or (saco_1==16) or (saco_1==24) or (saco_1==32) or (saco_1==40) or
(saco_1==48) or (saco_1==56) then
PTP P8 CONT Vel= 100 % PDAT8 Tool[1] Base[3]
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;permiso para sacar palet;;;;;;;;;;;;
wait sec 0
if (saco_1==sacos_1) then

```

```

pulse ($out[10],true,2)
saco_1=0
base_data[3]=base_data[3]
wait sec 0.5
endif

;;;;;;;;;;;;;incremento sacco;;;;;;;;;;

saco_1=saco_1+1

ptp_rel{z 110}c_ptp

;;;;;;;;;;;;;punto  elevado encima de zona 2

SYN PULSE 17 'Abrir_Manos' State= TRUE Time= 0.5 sec at START Delay= 0 ms

PTP P16 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

PTP P11 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P24 CONT Vel= 100 % PDAT24 Tool[1] Base[0]
PTP P25 CONT Vel= 100 % PDAT25 Tool[1] Base[0]

;verificacion manos y dedos abiertos
continue
loop
continue
if (($in[27]==false) or ($in[28]==false)) then
$timer_stop[1]=false
endif
continue
if $timer[1]>=4000 then
wait sec 0
$timer_stop[1]=true
$timer[1]=0
pulse ($out[12],true,2)
endif
continue
if (($in[27]==true) and ($in[28]==true)) then
$timer_stop[1]=true
$timer[1]=0
exit
endif
endloop
END

```

12.7 ZONA_111.src /ZONA_222.src.

DEF zona_111()

alt_incr_1=165
\$out[30]=true
coger_1 ()

;;;;;;;;saco 1 capa 1;;;;;;;;;;;;;

if (saco_1==1) or (saco_1==13) or (saco_1==25) then
PTP P1 CONT Vel= 100 % PDAT1 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse (\$out[19],true,2)
wait sec 0.3
endif

;;;;;;;;saco 1 capa 2;;;;;;;;;;;;;

Zona_112.src = Zona_222.src
DEF zona_111() = DEF zona_222 ().
Coger_1 () = Coger_2 ().
Saco_1 = sacco_2.
Base_data[3] = base_data[1].
Base_data[4] = base_data[2].
Sacos_1 = Sacos_2.
pulse_out[10] = pulse_out[11].

Zona_222.src
sería igual que
Zona_111.src
pero cambiando
las variables.

```
if (saco_1==4) or (saco_1==16) or (saco_1==28) then
PTP P4 CONT Vel= 100 % PDAT4 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif
```

;;;;;;;;saco 1 capa 3;;;;;;;;

```
if (saco_1==7) or (saco_1==19) or (saco_1==31) then
PTP P7 CONT Vel= 100 % PDAT7 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif
```

;;;;;;;;saco 1 capa 4;;;;;;;;

```
if (saco_1==10) or (saco_1==22) or (saco_1==34) then
PTP P10 CONT Vel= 100 % PDAT10 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif
```

;;;;;;;;saco 2 capa 1;;;;;;;;

```
if (saco_1==2) or (saco_1==14) or (saco_1==26) then
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif
```

;;;;;;;;saco 2 capa 2;;;;;;;;

```
if (saco_1==5) or (saco_1==17) or (saco_1==29) then
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif
```

;;;;;;;;saco 2 capa 3;;;;;;;;

```
if (saco_1==8) or (saco_1==20) or (saco_1==32) then
PTP P2 CONT Vel= 100 % PDAT2 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif
```

;;;;;;;;saco 2 capa 4;;;;;;;;

```
if (saco_1==11) or (saco_1==23) or (saco_1==35) then
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[3]
ptp_rel {z -140} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
endif
```

;;;;;;;;saco 3 capa 1;;;;;;;;

```
if (saco_1==3) or (saco_1==15) or (saco_1==27) then
PTP P3 CONT Vel= 100 % PDAT3 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
```

```
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 3 capa2;;;;;;;;;;;;

if (saco_1==6) or (saco_1==18) or (saco_1==30) then
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 3 capa3;;;;;;;;;;;;

if (saco_1==9) or (saco_1==21) or (saco_1==33) then
PTP P3 CONT Vel= 100 % PDAT3 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;saco 3 capa4;;;;;;;;;;;;

if (saco_1==12) or (saco_1==24) or (saco_1==36) then
PTP P6 CONT Vel= 100 % PDAT6 Tool[1] Base[3]
ptp_rel {z -100} c_ptp
pulse ($out[19],true,2)
wait sec 0.3
base_data[3].z=base_data[3].z+alt_incr_1+altura_1
endif

;;;;;;;;;;;;permiso para sacar palet;;;;;;;;;;;;
wait sec 0
if (saco_1==sacos_1) then
pulse ($out[10],true,2)
saco_1=0
base_data[3]=base_data[1]
wait sec 0.5
endif

;;;;;;;;;;;;incremento saco;;;;;;;;;;;;

saco_1=saco_1+1

ptp_rel{z 110}c_ptp

;;;;;;;;;;;;punto elevado encima de zona 2

SYN PULSE 17 'Abrir_Manos' State= TRUE Time= 0.5 sec at START Delay= 0 ms

PTP P16 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

PTP P11 CONT Vel= 100 % PDAT1 Tool[1] Base[0]
PTP P30 CONT Vel= 100 % PDAT30 Tool[1] Base[0]
PTP P31 CONT Vel= 100 % PDAT31 Tool[1] Base[0]

;verificacion manos y dedos abiertos
continue
loop
continue
if (($in[27]==false) or ($in[28]==false)) then
$timer_stop[1]=false
endif
continue
if $timer[1]>=4000 then
```



```
wait sec 0
$timer_stop[1]=true
$timer[1]=0
pulse ($out[12],true,2)
endif
continue
if (($in[27]==true) and ($in[28]==true)) then
$timer_stop[1]=true
$timer[1]=0
exit
endif
endloop
END
```

12.8 COGER_1.src

```
DEF coger_1( )

PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]
;punto de recogida
PTP P6 Vel= 80 % PDAT6 Tool[1] Base[0]
cerrar ()
wait sec 0.6
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]

PTP P8 CONT Vel= 100 % PDAT15 Tool[1] Base[0]
PTP P9 CONT Vel= 100 % PDAT16 Tool[1] Base[0]

END
```

12.9 COGER_2.src

```
DEF coger_2( )

PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]
;punto de recogida
PTP P6 Vel= 80 % PDAT6 Tool[1] Base[0]
cerrar ()
wait sec 0.6
PTP P5 CONT Vel= 100 % PDAT5 Tool[1] Base[0]

PTP P8 CONT Vel= 100 % PDAT15 Tool[1] Base[0]
PTP P11 CONT Vel= 100 % PDAT11 Tool[1] Base[0]
PTP P9 CONT Vel= 100 % PDAT16 Tool[1] Base[0]
PTP P10 CONT Vel= 100 % PDAT10 Tool[1] Base[0]

END
```

12.10 CERRAR.src

```
DEF cerrar( )

pulse ($out[18],true,2)
pulse ($out[20],true,2)

END
```